Diploma Thesis

– Florian Dreifus –

Aggregation of Large-Scale Network Flow Problems with Application to Evacuation Planning at SAP

Advisor:

Prof. Dr. Horst W. Hamacher

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

Vorwort

Wie, Du studierst Mathematik ? Was kann man denn damit später mal machen ? Eine Fragekonstellation, die ich in den letzten Jahren des Öfteren zu hören bekam. Viele Menschen assoziieren mit Mathematik, die schlimmsten Stunden ihrer Schulzeit. Was auch nicht sonderlich verwundert, denn allzu oft wird vergessen, die vielfältigen Anwendungsmöglichkeiten der Mathematik aufzuzeigen. Mathematik interessierte mich schon immer, jedoch war es mir schon zu Anfang meines Studiums wichtig, dass für das jeweilige Theoriegebiet auch eine Anwendungsmöglichkeit in der Realität existiert. Hier hatte ich das große Glück schon während meines Grundstudiums die Vorlesungen von Prof. Hamacher besuchen zu können. Er versteht es den Brückenschlag von Theorie und Anwendung, sei es durch Beispiele oder Übungsaufgaben, immer wieder herzustellen. Sein Lehrstuhl bietet eine Vielzahl von Arbeitsgebieten mit Bezug zur Praxis. Schon früh gab er mir die Möglichkeit mich als Hiwi an einem Projekt der Deutschen Forschungsgemeinschaft einzubringen. Im Mittelpunkt dieses Projekts stand die Entwicklung von Algorithmen für dynamische Netzwerkflüsse mit Anwendungen in der Evakuierungsplanung, wodurch meine Vertiefungsrichtung fast schon vorprogrammiert war. Bei der Auswahl meines Diplomarbeitsthemas, war es mir wichtig ein reales „Testobjekt“ zu finden, an dem ich die Theorie auch anwenden konnte. Diese Möglichkeit wurde mir durch das Facility Management der SAP AG gegeben. Im Rahmen eines Praktikums konnte ich Kontakt zu Herrn Seiberth herstellen, der mir neben seinem Know How auch den Bauplan des Erdgeschosses des Hauptgebäudes der SAP zur Verfügung stellte.
Gegeben war also nun der Bauplan eines Gebäudes und die Frage wie Mathematik die Evakuierungsplanung unterstützen kann ?
Daher beschäftigt sich der erste Teil der Diplomarbeit auch mit dem Thema wie ein gegebener Bauplan als Netzwerk modelliert werden kann. Nach der Modellierung des Netzwerkes stellt es kein Problem dar eine geeignete mathematische Formulierung zu finden, deren Ziel es ist die minimale Evakuierungszeit zu ermitteln (Evacuation Problem). Es zeigt sich jedoch recht schnell, dass die Problematik in der Lösung dieser Formulierung liegt und hier insbesondere bei der Größe der Netzwerke. Daher beschäftigt sich der zweite Teil der Diplomarbeit mit der Anwendung von Aggregation, zur Reduzierung der Netzwerkgröße. In einem ersten Schritt werden hierfür bestehende Erkenntnisse bezüglich der Aggregation des Transportation Problems und Minimum Cost Network Flow Problems vorgestellt. Die dort gesammelten Erkenntnisse werden in einem zweiten Schritt bezüglich ihrer Anwendungsmöglichkeit auf das Evacuation Problem untersucht. Hierfür konnten einige interessante Erkenntnisse abgeleiten werden.

Ich möchte dieses Vorwort auch dazu nutzen, mich bei einigen Menschen zu bedanken, die mich auf meinem bisherigen Weg begleitet haben und hoffentlich noch lange begleiten werden.
In erster Linie möchte ich meinen Eltern und meiner Oma danken. Sie geben mir moralische Unterstützung für mein Tun und stehen mir immer mit Rat und Tat zu Seite. Auch meiner Verlobten möchte ich großen Dank aussprechen, für ihr schier unerschöpfliches Verständnis, ihre Unterstützung und den Verzicht den sie in den letzten Jahren geübt hat. Ohne euch wäre vieles nicht möglich gewesen. Ich bin froh dass es euch gibt.
Ohne wahre Freunde im Leben ist man einsam und kann vieles nicht erreichen. Daher bin ich froh in Peter Bohrer solch einen wahren Freund gefunden zu haben, mit dem ich das

Mathematikstudium durchgangen bin. Neben vielem anderen, werde ich unsere Diskussion auf der täglichen Fahrt nach Kaiserslautern vermissen und wünsch ihm viel Erfolg für seinen neuen Lebensabschnitt. Auch möchte ich mich bei einem weiteren wahren Freund bedanken. Christian Traxel hat immer ein offenes Ohr für mich und gab mir Beistand wo immer er nur konnte. Auch ihm wünsche ich weiterhin viel Erfolg bei seinem Studium.
Mein besonderer Dank gilt auch Herrn Prof. Hamacher. Er gab mir schon früh die Möglichkeit an seinem Lehrstuhl zu arbeiten, prägte durch seinen Vorlesungsstil meine Arbeitsweise und unterstützte meine Bestrebungen bezüglich der Diplomarbeit. Auch möchte ich mich an dieser Stelle bei Stefan Ruzika für die Unterstützung meiner Diplomarbeit und die von ihm angebrachten Verbesserungsvorschläge bedanken.
Ohne die Bereitstellung des Bauplans und seines Wissens wäre meine Diplomarbeit nicht in dieser Form realisierbar gewesen, daher ein ganz großer Dank an Herrn Seiberth von der SAP.

Der Herr ist meine Kraft und mein Schild,
mein Herz vertraut ihm.
(AT, Psalm 28,7)

Lernen ist wie Rudern gegen den Strom.
Hört man damit auf, treibt man zurück.
(Laotse, chinesischer Philosoph)

Contents

Symbol Index

Networks

Dynamic

A	set of arcs
c_{ij}	cost for sending one unit flow on arc (i, j)
d	super sink
D	set of destination nodes
EV_i	supply of node i *(negative supply = demand)*
G_{DYN}	dynamic network
G_{STA}	static network, as a component of G_{DYN}
h_i	holdover flow capacity of node i
I	set of intermediate nodes
λ_{ij}	travel time along an arc (i, j)
N	set of nodes
S	set of sources
s	super source
T^*	optimal evacuation time
u_{ij}	flow capacity of arc (i, j)
$x(\bullet)$	dynamic flow
$x_{ij}(t)$	movement flow on arc (i, j) at time t
$x_{ii}(t)$	holdover flow at node i at time t
$x^*(\bullet)$	an optimal flow for the evacuation problem

Time Expanded

A^H	set of holdover arcs
A^M	set of movement arcs o
A^{TE}	set of arcs
d^{TE}	super sink
f	static flow
G_{TE}	time expanded network, corresponding to a dynamic network G_{DYN}
N^{TE}	set of nodes
s^{TE}	super source

Aggregation

Transportation Problem

Original Problem

a_s	supply of source s
b_s	(positive) demand of destination d
c_{sd}	cost for shipping one unit flow from source s to destination d
D	set of destinations
S	set of sources
x_{sd}	flow from source s to destination d
x^*	an optimal flow for the transportation problem
z^*	optimal objective value

Aggregated Problem

$\overline{a}_k$	supply of source k
$\overline{b}_l$	demand of destination l
$\overline{c}_{kl}$	cost for shipping one unit flow from source k to destination l
$\overline{D}$	set of destinations
$\overline{S}$	set of sources
$(\overline{u}, \overline{v})$	optimal dual pair
$\overline{y}_{kl}$	flow from source k to destination l
$\overline{y}^*$	an optimal flow for the problem
$\overline{z}^*$	optimal objective value

General Formulations

$\overline{DP}$	partition of the set of destinations D
$k(s)$	that index $k \in \overline{S}$ s.t $s \in S_k$, $s \in S$
$l(d)$	that index $l \in \overline{D}$ s.t. $d \in D_l$, $d \in D$
$\overline{SP}$	partition of the set of sources S

Minimum Cost Network Flow Problem

Original Problem

A	set of arcs

b_i	supply of node i (negative supply = demand)
c_{ij}	cost for sending one unit flow on arc (i, j)
D	set of destinations
I	set of intermediate nodes
l_{ij}	lower bound on the flow for arc (i, j)
N	set of nodes
S	set of sources
u_{ij}	flow capacity of arc (i, j)
x_{ij}	flow on arc (i, j)
x^*	optimal flow for the minimum cost network flow problem
$\overline{X}$	set of feasible solutions
z^*	optimal objective value

Aggregated Problem

$\overline{A}$	set of arcs
$\overline{b}_n$	supply of node n (negative supply = demand)
$\overline{c}_{np}$	cost for sending one unit flow on arc (n, p)
$\overline{D}$	set of destination
$\overline{I}$	set of intermediate nodes
$\overline{N}$	set of nodes
$(\overline{\pi}, \overline{\alpha})$	an optimal dual pair
$\overline{S}$	set of sources
$\overline{u}_{np}$	flow capacity on arc (n, p)
$\overline{y}_{np}$	flow on arc (n, p)
$\overline{y}^*$	an optimal flow for the aggregate minimum cost network flow problem
$\overline{Y}$	set of feasible solutions
$\overline{z}^*$	optimal objective value

General Formulations

$\overline{NP}$	partition of the node set
$n(i)$	that index $n \in \overline{N}$ s.t. $i \in J_n, i \in N$
$p(j)$	that index $p \in \overline{N}$ *s.t.* $j \in J_p$, $j \in N$

Evacuation Problem

Vertical Aggregation

$\overline{\lambda}_{ij}$	travel time along an arc (i, j)

$\overline{T}$	time horizon
$\overline{u}_{ij}$	flow capacity on arc (i, j)
$x_{ij}(\overline{t})$	movement flow on arc (i, j) at time $\overline{t}$
$x_{ii}(\overline{t})$	holdover flow at node i at time $\overline{t}$

Horizontal Aggregation

$\overline{A}$	set of arcs
$\overline{D}$	set of destinations
$\widetilde{EV}_n$	supply of node n *(negative supply = demand)*
$\tilde{h}_n$	holdover flow capacity of node n
$\tilde{\lambda}_{np}$	travel time along an arc (n, p)
$\overline{N}$	set of nodes
$\overline{S}$	set of sources
$\tilde{u}_{np}$	flow capacity on arc (i, j)
$\overline{y}_{np}(t)$	movement flow on arc (i, j) at time t
$\overline{y}_{nn}(t)$	holdover flow at node n at time t

General Functions

$\lvert\bullet\rvert$	cardinality if the expression is a set, modulus if the expression is a number
$[\![\bullet]\!]$	rounding an expression to the next integer
$pred(i)$	set of all predecessors of a node i
$succ(i)$	set of all successors of a node i

Set of Numbers

$\mathbb{R}_0^+$	set of nonnegative real numbers
$\mathbb{Z}_0^+$	set of nonnegative integer numbers

Chapter 1

Introduction

Our initial situation is as follows: The blueprint of the ground floor of SAP's main building the EVZ is given and the open question on how mathematic can support the evacuation's planning process ? Before we come to this question, we should define the term "evacuation".

Evacuation: *the act of evacuating; leaving a place in an orderly fashion; especially for protection* [Wor05]

There are a number of reasons why people need protection. In the case of the emergency evacuation of a building the threat of smoke and fire is perhaps the most obvious reason. Other reasons may include natural gas leaks, earthquakes or bomb threats. The evacuation process has to be well planned and defined in order to protect human beings in the emergency case. Therefore, architects, building designers and facility managers are mainly interested in two issues: How can large buildings with many occupants be evacuated in a minimum time and where are bottlenecks likely occurring in such an evacuation ?
In most cases regular practice evacuations are done to address these questions. They are done in order to make the occupants familiar with the evacuation procedure and to collect data about the evacuation process. However, the main drawbacks of practice evacuations lie in the fact that most of the occupants do not take them seriously as well as at the great expenses coming along with them. Also, they can only take place when a building has already been completed. But particularly in the planning phase of a building it is easy to make changes on the building characteristics if it is necessary (e.g. when bottlenecks are detected).
At this point of the discussion, we can come back to our initial questions on how mathematic can support the evacuation's planning process. To model evacuation processes in advance as well as for existing buildings two models can be used: *macro- and microscopic models*.
Microscopic models emphasize the individual movement of evacuees. These models consider individual parameters such as walking speed, reaction time or physical abilities as well as the interaction of evacuees during the evacuation process. Because of the fact that the microscopic model requires lots of data, simulations are taken for implementation. Most of the current approaches concerning simulation are based on cellular automats (e.g. [BA99]).
In contrast to microscopic models, *macroscopic models* do not consider individual parameters such as the physical abilities of the evacuees. This means that the evacuees are treated as a homogenous group for which only common characteristics are considered; an average human being is assumed. We do not have that much data as in the case of the microscopic models. Therefore, the macroscopic models are mainly based on optimization approaches. In most cases, a building or any other evacuation object is represented through a static network $G_{STA} = (N, A)$. A time horizon T is added, in order to be able to describe the evolution of the evacuation process over time. Connecting these two components we finally get a dynamic network $G_{DYN} = (N, A, T)$. Based on this network, dynamic network flow problems are formulated, which can map evacuation processes.

It is obvious that both models aim at the same objective. However, they use different approaches for reaching this objective. The following figure shows that both models could be used to get better solutions for the evacuation planning. Due to the input parameters, the macroscopic modeling should yield a lower bound for the evacuation process whereas the microscopic modeling should yield an upper bound. This interrelation between both approaches has to be validated.

Figure 1: Sandwich Approach (to be validated)

In our work, we focused on the macroscopic model, using the dynamic network flow approach. The following figure summarizes the statements made so far and gives a first impression about the structure of our thesis.

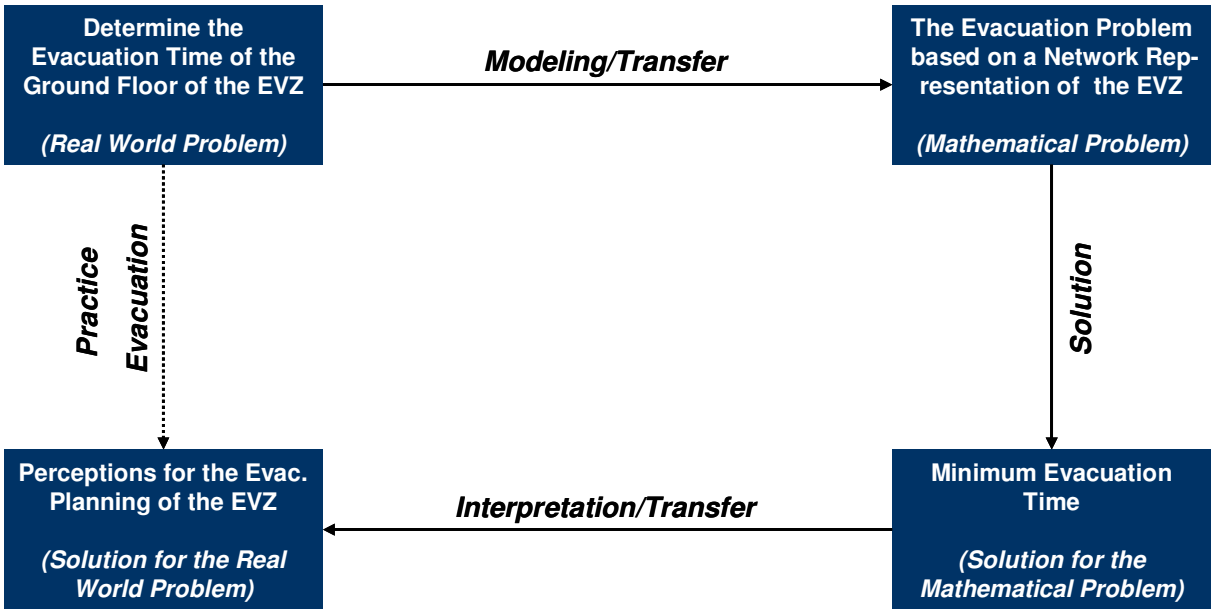


Figure 2: Using mathematical approaches for solving real world problems

In this thesis, we will address two of the questions coming along with Figure 2: Given the blueprint of the EVZ, how can we transfer the real world problem into the mathematical world ? And, how can we solve the mathematical problem after the transfer ?
Our main focus concerning the transfer from the real world problem will be the modeling of the blueprint as a dynamic network. After modeling the blueprint as a dynamic network, it will be no problem to give a formulation of a dynamic network flow problem, the so-called *evacuation problem*, which seeks for an optimal evacuation time. However, we have to solve a static large-scale network flow problem to derive a solution for this formulation. In order to reduce the network size, we will examine the possibility of applying aggregation to the evacuation problem.
Aggregation (*lat. aggregare = piling, affiliate; lat. aggregatio = accumulation, union; the act of gathering something together)* was basically used to reduce the size of general large-scale linear or integer programs (see [EPRW91] for a detailed overview). The results gained for the general problem definitions were then applied to the transportation problem and the minimum cost network flow problem. We review this theory in detail and look on how results derived there can be used for the evacuation problem, too. As the following figures show the theory can be divided into two parts, depending on the kind of solution provided for the original problem (feasible vs. optimal).

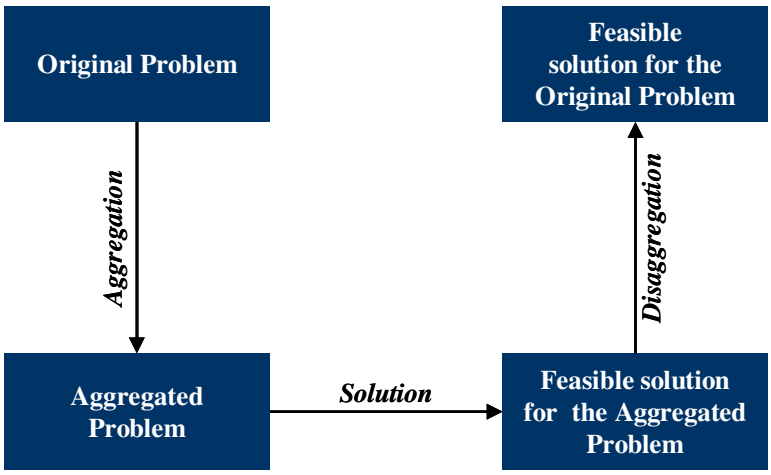


Figure 3: Derive a feasible solution by using aggregation

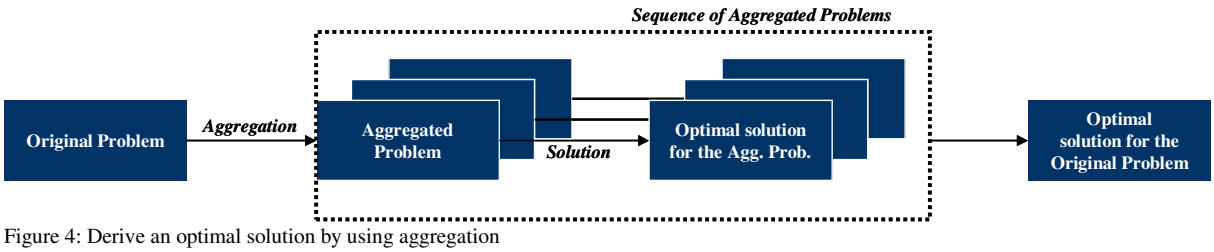


Figure 4: Derive an optimal solution by using aggregation

As mentioned before we discuss the working fields of modeling the real world problem and solving the corresponding mathematical problem (see Figure 1). The interpretation and transfer of the results into the real word is omitted in our work.

In the following, we provide an overview about how our thesis is organized.
In **Chapter 2**, we give the formulation of the evacuation problem, which is based on the quickest flow problem. In addition, well known dynamic network flow problems, such as the earliest arrival flow problem, are reviewed. They can also be used for modeling evacuation processes. Moreover, we discuss the possibility of representing dynamic network flow problems as static problems, in the *time expanded network*. In order to model the building as a dynamic network, the concepts of Fruin [Fru71] concerning the derivation of parameters are reviewed in **Chapter 3**. His theory is applied for modeling the ground floor of SAP's main building, the EVZ. Modeling the ground floor will show the necessity of reducing the network size in order to solve the evacuation problem, especially if we want to model the complete building. Therefore, **Chapter 4** starts with a review of aggregation theory applied to the transportation problem. Here, we mainly focused our discussion on the work of Balas [Bal65] and Zipkin [Zip80]. Balas derived an algorithm which uses aggregation and finally leads to an optimal solution. In contrast to Balas, Zipkin designed in his concepts (*weighted aggregation*) a specific disaggregation mapping which finally leads to a feasible solution for the original problem. Aggregation applied to the more general minimum cost network flow problem will be discussed in **Chapter 5**. The basic concepts for aggregation are the same as for the transportation problem. We review the concepts of Lee [Lee75], Francis [Fra85] and Zipkin [Zip80]. For the approach of Lee and Francis based on the concepts of Balas (*aggregation by dominance*) we will be able to present a bound on the loss of accuracy introduced by solving the aggregated problem instead of the original one. In **Chapter 6**, we will have a look at the possibilities of applying the results of aggregation gained for the transportation problem and the minimum cost network flow problem to the evacuation problem. The discussion will be separated into two parts, depending on whether we apply the aggregation on the horizontal (i.e. reduction of the time horizon) or vertical dimensions (i.e. grouping of original nodes). We are able to show that the optimal evacuation time of the aggregated problem is equal to the optimal time of the original one, for a special case of aggregation. Chapter 6 will be closed with some empirical tests about the impact of aggregation applied to the evacuation problem. In **Chapter 7**, we conclude our discussion with some ideas for possible future research.

Chapter 2

Modeling Evacuation Processes with Dynamic Network Flow Problems

The following chapter gives an overview about how the concepts of dynamic network flow problems (DNFP) can be used in order to model evacuation processes. Dynamic network flow problems are an extension of the static setting. They are based on a dynamic network $G_{DYN} = (N, A, T)$ which consists of a static network $G_{STA} = (N, A)$ and a time horizon T. Each arc and node has particular parameters, such as a capacity. Depending on whether a discrete or a continuous representation of time is used, dynamic network flow problems can be formulated in two ways. In a *continuous* dynamic network flow model the parameter T is treated as a real number. This means that the flow is distributed continuously over time. In our work, the focus lies on *discrete* time dynamic network flow problems. In this case, the flow is distributed over a set of predetermined time units $t = 0,1,...,T$.
Our main focus will be on the formulation of the evacuation problem. In the evacuation problem we look for a flow which minimizes the total evacuation time, i.e. the time needed to evacuate (clear) a building. In order to map the evacuation process, the static network G_{STA} will be used to model supply and demand points as well as routes which can be used to transfer supplies to demands. These routes may have some intermediate transshipment points which have neither supply nor demand. The supply points are modeled as nodes and can be interpreted as locations (e.g. rooms, offices, lobbies) of evacuees in the beginning of the evacuation. Besides the transshipment points, the demand points are also modeled as nodes. Demand points can be interpreted as safety areas. The routes an evacuee can take are modeled by paths of the graph. A path consists of nodes and arcs, where an arc connects two adjacent nodes. The time horizon T, the second component of a dynamic network, is needed for two reasons: The fist reason is that the evolution of the flow over time can not be expressed by a static network. The second reason concerns the modeling of parameters. If we want to model a blocked hallway (e.g. blocked by fire or smoke) for instance, we should have the possibility to change parameters temporally. In the case of a blocked hallway, we would set the capacity equal to zero after a particular time unit. In the same way, such settings can not be realized within a static problem formulation.

In the following we provide an overview about discrete-time dynamic network flow problems with time independent parameters. In the first section, we will formulate the evacuation problem. Section 2.2 gives an overview about further problem formulations concerning evacuation processes and shows a very interesting interrelation between them. The chapter will be concluded with Section 2.3 in which we describe the *time expanded network*, which is the equivalent static representation of a dynamic network.

2.1 The Evacuation Problem

A discrete-time dynamic network $G_{DYN} = (N, A, T)$ consists of a static network $G_{STA} = (N, A)$ with the set of nodes *N* and the set of arcs *A*, a finite time horizon $T \in \mathbb{Z}_0^+$ and corresponding parameters. The time horizon is discretized into the set $\{0,1,...,t,...,T\}$.

For each node $i \in N$ we have a particular demand/supply denoted with EV_i, where we can distinguish three different cases:

$$\mathrm{EV_i} \begin{cases} > 0, \text{ node } i \text{ is a (supply node) source} \\ < 0, \text{ node } i \text{ is a (demand node) sink} \\ = 0, \text{ node } i \text{ is a transshipment node} \end{cases}$$

Corresponding to EV_i we can partition *N* into the following sets:

$$\mathrm{S} = \{\mathrm{i} \in \mathrm{N} : \mathrm{EV_i} > 0\}$$
$$\mathrm{D} = \{\mathrm{i} \in \mathrm{N} : \mathrm{EV_i} < 0\}$$
$$\mathrm{I} = \{\mathrm{i} \in \mathrm{N} : \mathrm{EV_i} = 0\}$$

Each node $i \in N$ has a node capacity $h_i \in \mathbb{R}_0^+$. The node capacity or, equivalently, the holdover capacity defines the maximum amount of flow that can be held over one time unit at node *i*. The demand/supply EV_i and the holdover capacity h_i describe node attributes. In contrast to the holdover capacity the demand/supply is well known from static problems.
We go on with the attributes of arcs $(i, j) \in A$. For each arc $(i, j) \in A$ we have a travel time $\lambda_{ij} \in \mathbb{Z}_0^+$, a capacity $u_{ij} \in \mathbb{R}_0^+$ and, depending on the problem formulation, cost $c_{ij} \in \mathbb{R}$. In contrast to the interpretation of capacity in the dynamic case, the parameter cost has the same interpretation as for static problems. The capacity of an arc $(i, j) \in A$ does not bound the total flow on that arc at a given time *t*, but defines the maximum number of flow that can enter the arc at each time *t*. For example an arc with capacity three and travel time two can accept three new units of flow at each time step, for a total of up to six units of flow in transit on this arc at one time unit.
As mentioned before, the time horizon *T* is discretized into the set $\{0,1,...,t,...,T\}$. The number of time periods *T* depends on the basic time unit π in which travel times are measured (i.e. we get *T* by dividing the planning horizon of interest by π and rounding up the result). If we set the basic time unit equal to two for instance (i.e. $\pi = 2$), then specify five time units for passing an arc $(i, j) \in A$ means we need ten seconds to do so. The closer π is to one (i.e. one time period corresponds to one second) the more accurately the model represents the actual flow evolution. Choosing π too small, however, will result in undesirable size of the network and can lead to dynamic capacities that are not integer. This can violate an all-integer requirement of whatever algorithm is used to solve the problem. As a result, the choice of π is a compromise between model realism and model complexity.
In order to get a complete model of any dynamic network flow problem, one important definition has left so far, the definition of a dynamic flow. Before we come to this definition we make an important remark about the parameters of nodes and arcs defined by now.

Remark: For our definition of the evacuation problem we assumed time-independent parameters, a special case of the more general formulation, in which time dependent parameters are allowed. In this case, the parameters are defined for each time unit $t = 0,1,...,T$. Hence, they can change over time. Of course, the adoption of the more general model would lead to more possibilities concerning the modeling of evacuation processes (e.g. arcs representing hallways which can not be passed after a particular time unit caused by smoke). In our paper, however, it is sufficient to use time independent parameters.

We continue with the definition of a dynamic flow.

Definition 2.1

A dynamic flow *x* over the time horizon *T* is given by the following mapping:

$$x : \left(A \cup \{(i,i) : i \in N\}\right) \times \{0,...,T\} \longrightarrow \mathbb{R}_0^+$$

Here, we assume for notational convenience throughout our work that $x_{ij}(t) = 0$ for $t < 0$, $(i, j) \in A$ as well as $x_{ii}(t) = 0$ for $t < 0$, $i \in N$.

$x_{ij}(t)$ determines the flow entering arc (i, j) at time unit $t \in \{0,1,...,T\}$. Hence, the flow leaves the arc at $t' = t + \lambda_{ij}$. The amount of holdover flow at node *i* at time unit *t* is represented by $x_{ii}(t)$. Holdover means that the flow stays for one time unit (i.e. $\lambda_{ii} = 1$) at node *i*.
Like in the static case we introduce a super source *s* and super sink *d*, because it is possible that $|S| > 1$ or $|D| > 1$. By introducing a super source and a super sink the set of nodes and arcs has to be adjusted. This means that we have to add *s* and *d* to the set of nodes *N* as well as arcs (s,i) with $i \in S$ and (i,d) with $i \in D$ to the set of arcs *A*.
We set the supply for the super source to $\sum_{i \in S} EV_i$ and the demand of the super sink to $\sum_{i \in D} EV_i$. The travel time of arcs $(i,d), i \in D$ is set to zero, whereas the capacity is set equal infinity. For arcs between *s* and *i*, with $i \in S$, the travel time is also zero, but the capacity is set to EV_i.

In order to derive a dynamic network flow model for the evacuation problem we take the model of Hamacher and Tjandra [HT01]. They adapted the quickest flow problem in order to find the minimum evacuation time for a given number of building occupants. In general, the quickest flow problem seeks for a dynamic flow which sends a given amount of flow *v* from a source *s* to a sink *d* in minimum time (e.g. see Burkard, Dlaska and Klinz [BDK93] or Fleischer and Tardos [FT98]).

Evacuation Problem

(EVAC)

$$\min T$$

s.t.

$$x_{ii}(t-1) - x_{ii}(t) = \sum_{j:(i,j)\in A} x_{ij}(t) - \sum_{j:(j,i)\in A} x_{ji}(t-\lambda_{ji}) \qquad (2.1)$$

$$t = 0,1,...,T;\ \forall i \in N \setminus \{s,d\}$$

$$\sum_{t=0}^{T} \sum_{i\in D} x_{id}(t) = \sum_{i\in S} EV_i \qquad (2.2)$$

$$x_{si}(0) = EV_i, \qquad \forall i \in S \qquad (2.3)$$

$$x_{ii}(t) = 0, \qquad \forall i \in D;\ t = 0,1,...T \qquad (2.4)$$

$$0 \le x_{ii}(t) \le h_i, \qquad t = 0,1,...,T;\ i \in N \setminus D \qquad (2.5)$$

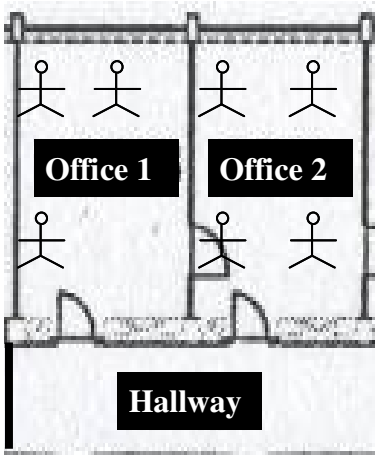
$$0 \le x_{ij}(t) \le u_{ij}, \qquad t = 0,1,...,T-\lambda_{ij};\ \forall (i,j) \in A \qquad (2.6)$$

Let us denote with x^* an optimal flow and with T^* the corresponding optimal evacuation time.

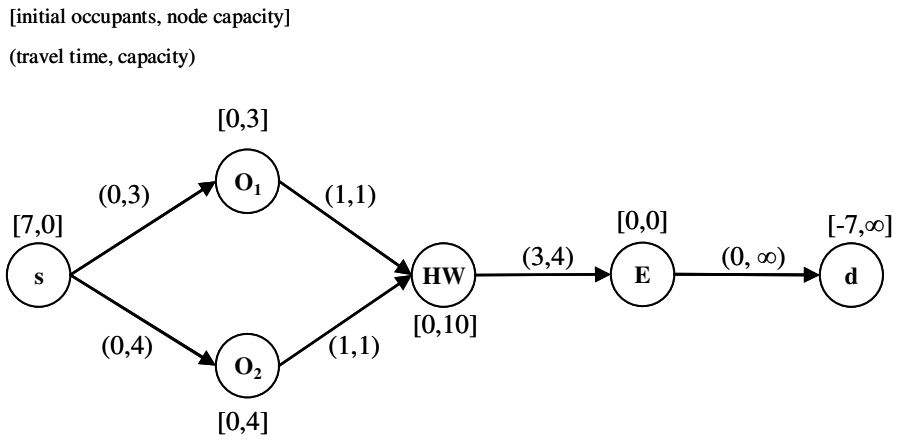
Constraint (2.1) is the dynamic flow conservation constraint. Constraints (2.2) and (2.3) assure that the right amount of flow reaches the super sink and leaves the super source, respectively. Because of the fact that we assume no holdover arcs for sink nodes we have introduced constraint (2.4). The protection of the particular node and arc capacities is done by constraints (2.5) and (2.6) respectively. The evacuation problem seeks for a flow which satisfies the constraints mentioned above in a minimal time *T*, indicated by the objective function.
General discrete-time dynamic network flow problems (DNFP) contain at least constraints (2.1), (2.5) and (2.6).
When we talk about static network flow problems we know that there exists an integral optimal flow as long as all capacities as well as the supplies/demands are integral and a feasible solution exists. This result holds for the dynamic case, too (e.g. see [Tja03]).
We conclude this section with a detailed example for the definition and formulations made so far.

Example 2.1

Blueprint of the building:



Network Representation:



The travel time is measured in seconds (i.e. $\pi = 1$)

The following flow yields an optimal solution of $T^* = 7$ seconds for the evacuation problem:

$x_{sO_1}(0) = 3; \quad x_{sO_2}(0) = 4;$

$x_{O_1HW}(0) = 1; \quad x_{O_1HW}(1) = 1; \quad x_{O_1HW}(2) = 1;$
$x_{O_1O_1}(0) = 2; \quad x_{O_1O_1}(1) = 1;$

$x_{O_2HW}(0) = 1; \quad x_{O_2HW}(1) = 1; \quad x_{O_2HW}(2) = 1; \quad x_{O_2HW}(3) = 1;$
$x_{O_2O_2}(0) = 3; \quad x_{O_2O_2}(1) = 2; \quad x_{O_2O_2}(2) = 1;$

$x_{HWE}(2) = 3; \quad x_{HWE}(3) = 1; \quad x_{HWE}(4) = 3;$
$x_{HWHW}(0) = 0; \; x_{HWHW}(1) = 2; \; x_{HWHW}(2) = 1; \; x_{HWHW}(3) = 2;$

$x_{Ed}(5) = 3; \quad x_{Ed}(6) = 1; \quad x_{Ed}(7) = 3;$

2.2 Further Dynamic Network Flow Problems

In the previous section, we presented a dynamic network flow model for calculating the minimum evacuation time. In the following we give an overview about further models which can be used for the computation of important values of an evacuation process.

2.2.1 The Average Evacuation Time Flow Problem

The objective of the evacuation problem defined in the previous section is the minimization of the time needed to clear a building. A very similar problem is the computation of the average evacuation time, i.e. the average time required by an evacuee to leave the building. In that case, a feasible flow has to satisfy the same constraints as in the case of the evacuation problem, but has to minimize a different objective function. We have to introduce the so called *turnstile costs* [CFS82] first, before we are able to define this new objective function. The turnstile costs measure the time when evacuees reach their final destination and can be defined as follows

Definition 2.2

$$c_{ij}(t) = \begin{cases} t, & i \in D \text{ and } j = d; \\ 0, & \text{else} \end{cases} \qquad \forall (i,j) \in A$$

Using the *turnstile costs* we can define the average evacuation time an evacuee needs to leave the building as follows

$$AET(x) = \frac{\sum_{t=0}^{T} \sum_{i \in D} c_{id}(t) x_{id}(t)}{EV_s} = \frac{\sum_{t=0}^{T} \sum_{i \in D} t x_{id}(t)}{EV_s}$$

We can see that the average evacuation time only depends on the flow x, since the total supply EV_s in the denominator is constant. Therefore, it is sufficient to minimize the numerator. Due to the similarity with the evacuation problem the dynamic network flow model which minimizes the average evacuation time can be formulated as follows:

Average Evacuation Time Flow Problem

(AETF)

$$\min \sum_{t=0}^{T} \sum_{i \in D} t x_{id}(t)$$

s.t.

$(2.1) - (2.6)$

2.2.2 Maximum Dynamic Flow and Earliest Arrival Flow Problems

In the case of the evacuation problem we have a predefined number of evacuees. This implies that we know the number of people located in the different areas. This seems to be no problem, if we model a building with restrictive access, such as a company building. However, it is more difficult to estimate the number of people in the different areas, modeling buildings with a large amount of visitors, like public buildings for instance. In such a case it is advisable to model the evacuation process as a maximum dynamic network flow problem (MDF) or as an earliest arrival flow problem (EAF). Both problems can be taken for modeling evacuation processes in which we have no reliable information about the number of occupants in the beginning of the evacuation. So let us come to the maximum dynamic flow problem first (see for example [FF58],[FF62]).
The objective of the MDF is to maximize the dynamic flow arriving at the sink for a given time horizon *T*. This means for the evacuation process that we are interested in the maximum number of people which can be evacuated in a given time horizon *T*. In contrast to the evacuation problem, the arcs leaving the super sink have infinite capacity. We further assume that there are no holdover arcs for all $i \in S$. Taking these assumptions into account the MDF can be formulated as follows:

Maximum Dynamic Flow Problem

(MDF)

$$\max \sum_{t=0}^{t=T} \sum_{i \in D} x_{id}(t)$$

s.t.

$$x_{ii}(t-1) - x_{ii}(t) = \sum_{j:(i,j) \in A} x_{ij}(t) - \sum_{j:(j,i) \in A} x_{ji}(t-\lambda_{ji}) \qquad (2.7)$$

$$t = 0,1,...,T; \forall i \in N \setminus \{s,d\}$$

$$x_{ii}(t) = 0, \qquad \forall i \in S \cup D; t = 0,1,...T \qquad (2.8)$$

$$0 \le x_{ii}(t) \le h_i, \qquad t = 0,1,...,T; i \in N \setminus S \cup D \qquad (2.9)$$

$$0 \le x_{ij}(t) \le u_{ij}, \qquad t = 0,1,...,T-\lambda_{ij}; \forall (i,j) \in A \qquad (2.10)$$

An optimal solution for the MDF defined above can be computed very easily by using the concepts of *time repeated flow* (TRF). By using the TRF approach, we have to solve a *minimum cost circulation problem* in the static network $G_{STA} = (N, A)$. Therefore we add an arc, to connect the super sink with the super source, which has cost equal to $-(T+1)$ and a flow capacity of infinity. The costs of the remaining arcs of G_{STA} are set equal to the travel times of the dynamic problem. The minimum circulation flow calculated for the static problem is decomposed into chain flows P_i, which start at the super source and end at the super sink. Each chain flow P_i is repeated from time zero till time $(T+1)-\lambda(P_i)$ in the dynamic network, where $\lambda(P_i)$ is the travel time of chain flow P_i. This approach can only be applied in the case of time independent parameters. For more details about this approach we suggest the work of Ford and Fulkerson [FF58], [FF62].

The EAF is a variant of the maximum dynamic flow problem. The only difference between both models lies in the objective function. The MDF seeks for a flow which sends the maximal number of units to the sink for a given time horizon *T*. In the case of the EAF we look for maximum dynamic flows reaching the sink at every time period $t = 0, 1, ..., T$. The relation between both problems is obvious: The optimal solution of the EAF is also a solution of the MDF; not only for the time horizon *T*, but also for any smaller time horizon $t \leq T$. This means for the evacuation process that we try to evacuate the maximum number of evacuees not only for the time horizon *T*, but also for any time unit smaller than *T*. Hence, the evacuation process modeled by the EAF is a safer and more realistic one. Because of the fact that we only have to change the objective function the EAF can be formulated as follows:

Earliest Arrival Flow Problem

(EAF)

$$\max \sum_{t=0}^{t=T'} \sum_{i \in D} x_{id}(t) \qquad \forall T' = 0,1,...,T$$

s.t.

$$(2.7) - (2.10)$$

According to the definition it is obvious that every earliest arrival flow is a maximum dynamic flow, whereas the reverse does not hold. In order to get more detailed information about the EAF we recommend the work of Hoppe and Tardos [HT94] and Gale [Gal59]. The latter introduced this problem, calling it the *Universal Maximum Flow Problem.*

2.2.3 The Triple Optimization Theorem

We want to conclude this section with a very interesting result about the interrelation of the problem formulations made so far. Until now we have seen three (four) objective functions which are important for evacuation processes.

- Minimization of the total time *T*; needed to evacuate the building
- Minimization of the average time; needed to clear the building
- Maximization of the output of evacuees for the first *t* periods, $t \leq T$

In general we are contented to be able to maximize or minimize a single objective function of our choice. However, Jarvis and Ratliff [JR82] showed that we are able to satisfy all three objective functions defined so far at the same time.

Theorem 2.1 (Triple Optimization Theorem of Jarvis and Ratliff [JR82])

Let us denote with F_t the flow vector of arcs connected to the super sink at time *t*. Let further C_t be the associate cost vector where we have that $c_1 < c_2 < ... < c_t$. Any feasible flow of *K* units from *s* to *d* that satisfies either condition I or II defined below also satisfies the other two

$$\text{I.)} \quad \max \sum_{t=1}^{T'} F_t \quad \forall T' \le T$$

$$\text{II.)} \quad \min \sum_{t=0}^{T} C_t F_t$$

$$\text{III.)} \quad \min\{T \mid F_{T'} = 0, \forall T' > T\}$$

It is enough to solve either I or II in order to get a solution for the other two problems, as well. This means that solving EAF or the AETF also yields a solution for the evacuation problem.

2.3 The Time Expanded Network

In the previous section, we have defined the EAF, the AETF and the evacuation problem that is based on the quickest flow problem. The computation of an optimal solution for these dynamic network flow problems is more complicated than for the static ones in general, because of the fact that the flow conservation constraint in the dynamic case is more complex than in the static case. The easy way of computing a solution for the MDF was an exception, resulting from our assumption of time independent parameters.
If we have a single source evacuation problem (i.e. we do not have to introduce a super source) the problem can be solved by using an interrelation with the maximum dynamic flow problem (see [BDK93]). But deriving an optimal solution for a multiple source evacuation problem things get more complicated.
However, it is possible to resolve these complications somehow, since every dynamic network flow problem can be transformed into an equivalent static network flow problem. There is an one-to-one correspondence between the flow of the static problem and the one of the dynamic problem. Therefore, it is possible to use well known network flow algorithms to solve the static problem and to map this solution to an optimal solution of the dynamic network flow problem.
We have seen in the last section that a dynamic network $G_{DYN} = (N, A, T)$ consists of a static network $G_{STA} = (N, A)$ and a time horizon *T*. In order to derive the so-called *time expanded network* $G_{TE} = (N_{TE}, A_{TE})$, each node $i \in N$ is copied $T+1$ times. Between two consecutive time copies $i(t)$ and $i(t+1)$ of a node $i \in N$, we have a *holdover arc* whose capacity is equal to the holdover capacity of *i*. For each arc $(i, j) \in A$ with travel time λ_{ij} and for every *t* between 0 and $T - \lambda_{ij}$, we have a *movement arc* in the time expanded network from time copy *t* of node *i* to time copy $t + \lambda_{ij}$ of node *j*. As mentioned before, every static flow *f* in G_{TE} has a corresponding dynamic flow *x* in G_{DYN}, and vice versa, where it is obvious that we get this one-to-one correspondence by

$$x_{ij}(t) = f_{i(t),j(t+\lambda_{ij})}$$

So let us state the comments made so far in a more formal way. Our definition is based on the concepts of Ford and Fulkerson [FF58].

Definition 2.3

The *time expanded network* $G_{TE} = (N_{TE}, A_{TE})$ which corresponds to a dynamic network $G_{DYN} = (N, A, T)$ consists of the set of nodes

$$N_{TE} = \{i(t) : i \in N \setminus \{s, d\}, t = 0, 1, ..., T\}$$

and the set of arcs A_{TE} which can be divided (i.e. $A_{TE} = A^M \cup A^H$) into the following two sets:

$$A^M = \{(i(t), j(t')) : (i, j) \in A; t = 0, 1, ..., T - \lambda_{ij}, t' = t + \lambda_{ij}\} \quad \textit{(Movement Arcs)}$$

$$A^H = \{(i(t), i(t+1)) : i \in N \setminus \{s, d\}; t = 0, 1, ..., T - 1\} \quad \textit{(Holdover Arcs)}$$

The capacity u_{ii} of the holdover arc $(i(t), i(t+1))$ is determined by the holdover capacity h_i defined for the dynamic problem. The capacity u_{ij} of the movement arc $(i(t), j(t'))$ is determined by the capacity u_{ij} of the arc $(i, j) \in A$.

Even though it is possible that the dynamic network flow problem is only a single source, single sink problem the corresponding time expanded network may have several sources and several sinks. Therefore it necessary to introduce a super source s^{TE} and a super sink d^{TE} to create a single source, single sink problem in the network G_{TE}. It depends on the underlying problem in which way the super source is connected to the time copies of the source nodes. In the case of the evacuation problem, the super source is only connected to the time zero copies of the source nodes. Therefore, the time copies of a source node are connected through holdover arcs. In the case of the MDF or the EAF there are no holdover arcs between time copies of a source node, since the super source is connected to all time copies of source nodes. Regarding the connection of sinks to the super sink, we have the same proceeding for every problem. In general, all time copies of every sink node are connected to the super sink and there are no holdover arcs between the time copies of a sink node.

We saw in the triple optimization theorem that it is sufficient to solve the AETF in order to get an optimal solution for the evacuation problem. In the following example we show how the AETF can be formulated as a static minimum cost network flow problem in the time expanded network. The AETF taken in the example is based on the evacuation problem presented in Example 2.1.

Example 2.2

The following optimal flow for the evacuation problem is derived by the optimal flow of the minimum cost network flow problem in G_{TE}:

$x_{sO_1}(0) = 3; \quad x_{sO_2}(0) = 4;$

$x_{O_1HW}(0) = 1; \quad x_{O_1HW}(1) = 1; \quad x_{O_1HW}(2) = 1;$

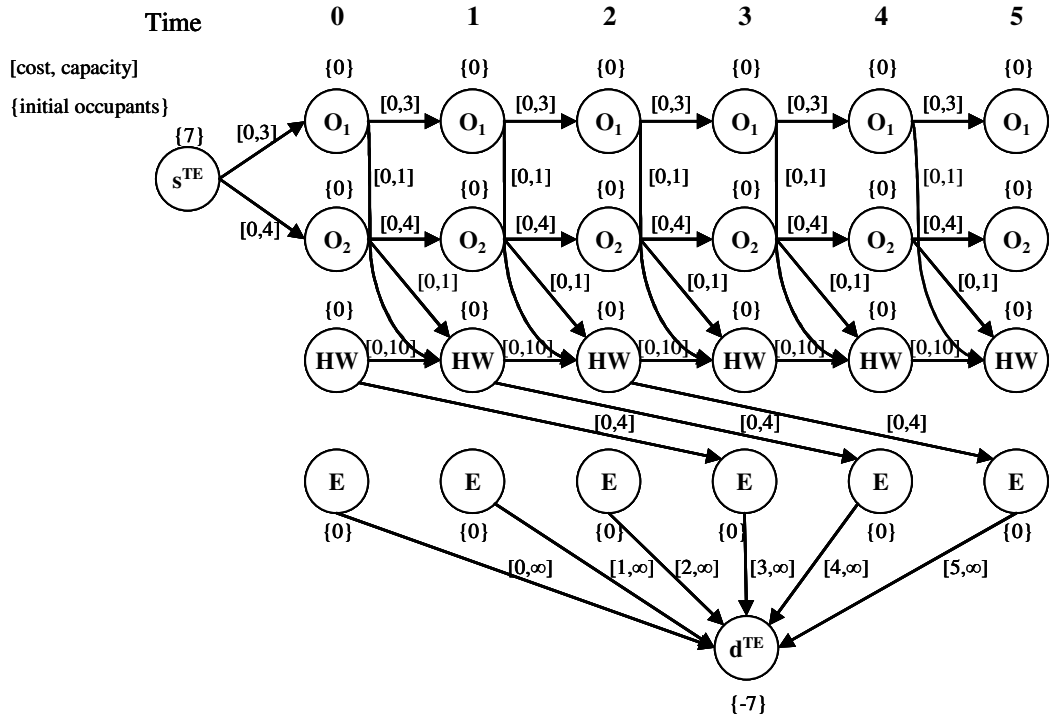
$x_{O_1O_1}(0) = 2; \quad x_{O_1O_1}(1) = 1;$

$$x_{O_2HW}(0)=1; \quad x_{O_2HW}(1)=1; \quad x_{O_2HW}(2)=1; \quad x_{O_2HW}(3)=1;$$
$$x_{O_2O_2}(0)=3; \quad x_{O_2O_2}(1)=2; \quad x_{O_2O_2}(2)=1;$$

$$x_{HWE}(1)=2; \quad x_{HWE}(2)=2; \quad x_{HWE}(3)=2; \quad x_{HWE}(4)=1;$$

$$x_{Ed}(4)=2; \quad x_{Ed}(5)=2; \quad x_{Ed}(6)=2; \quad x_{Ed}(7)=1;$$

Time Expanded Network G_{TE}:



As we have seen in the example above, the evacuation problem can always be solved as a static network flow problem in the larger time expanded network. So it does not seem to be necessary to have additional algorithms for solving the evacuation problem. However, if T is large, then the time expanded network becomes very large, since we copy each node $i \in N$ for each time unit $t = 0,1,...,T$. In the end, the number of computations needed to solve the evacuation problem using the time expanded representation becomes also very large. The dependence of the network size on the time horizon T is the main drawback of this approach. Algorithms for solving dynamic network flow problems which are based on the time expanded network have at most running times depending polynomially on T; such algorithms are *pseudo polynomial*.

Since we do not use any arc in the path from the super source to any sink node at time greater than T, we can reduce the size of the time expanded network by eliminating inessential arcs including the corresponding nodes (e.g. node *HW(3)*, *HW(4)* and *HW(5)* in the example above; for more details see [Tja03]). However, even though if we delete these nodes and arcs the size of the time expanded network remains one of the major problems concerning the solution of dynamic problems. Therefore, we will have a look on how aggregation theory can be used to reduce the size of the time expanded network. Before we come to sections on aggregation theory, we will have in the following a chapter about the modeling of buildings as networks. Modeling is an important step, because we will need the network representation to apply our dynamic network flow models.

Chapter 3

Modeling Evacuation Objects

In the following we will provide a brief guide regarding the modeling of evacuation objects as networks, focusing on the modeling of buildings. The theoretical modeling results will be applied to the ground floor of SAP's main building, the EVZ.
We have seen in the last chapter that if we have a representation of an evacuation object as a dynamic network, we are able to use algorithms to calculate values such as the minimum time required for clearing the building. Generally spoken rooms, hallways or locations in general can be modeled as nodes. The connection between these locations can be represented through arcs. After modeling a building, we have a static network representation $G_{STA} = (N, A)$. We have to introduce a time horizon T in order to map the evolution of the evacuation process over time. So we finally get the dynamic representation $G_{DYN} = (N, A, T)$.
The first modeling question we are faced with is to decide which locations should be represented through nodes. We will see that the question can be often answered by the given building characteristics. However, sometimes we have to make explicit assumptions.
The second problem lies in the definition of the parameters, such as the time required to travel from one location to another one. Whenever possible and practicable, it is best for the design to obtain real problem data, since the data depends on the specific building and occupants involved. In practice, however, neither the building exists already nor is the time available in most cases to observe all the data required. In addition it is nearly impossible to get realistic values, because in evacuation exercises most of the evacuees do not take the exercise seriously and in real emergency cases you are faced with other problems than collecting data.

We also have tried to get data by observing the flow of people on particular hallways of the EVZ. We counted the flow at three different day times. Between 8 and 9 pm, the time when most people start their work; at lunch time and between 4 and 5 am, when most people finish their work. Due to the factors mentioned above, it was not possible to derive realistic parameters needed for our model. Therefore we used the concepts of Fruin [Fru71] to derive the parameters required for modeling the ground floor of the EVZ.
Fruin's work has become the standard approach for many subsequent building design and planning operations. He defined the so-called "level of service concept" (LoS). In this concept the density and speed relationship are used as guidelines for comfort and safety. Fruin`s concepts are based on measurements and observations made in a pedestrian street environment. Nevertheless we will use them for defining flow attributes in buildings, too.

The concepts of Fruin are from the early seventies. Current work concerning the modeling of pedestrian flows can be found in the Ph. D. thesis of G. Keith Still [Sti00], for instance.
G. Keith Still reviewed in his Ph. D. thesis the ideas of Fruin and observed situations which contradict the assumptions of Fruin. He developed his own concepts for the modeling of pedestrian flow based on observations in stadiums. Because the focus of our work lies on the aggregation of large network flow problems, we will not discuss the concepts and drawbacks

of Fruin in detail, even though this is a very interesting field of research with a high degree of multidisciplinarity. We will also omit the modeling of pedestrian flow in staircases, because we are not faced with this kind of modeling in our real world problem instance. We recommend the work of Jake Pauls (e.g. [Pau78], [Pau82]) for those who are interested in very detailed information concerning the modeling of pedestrian flow in staircases.

The chapter is separated into three sections. We will start with a detailed look on how the evacuation specific characteristics of a building can be modeled as components of a network. The theoretical observation we made in Section 3.1 will be applied in Section 3.2 to model the ground floor of SAP' main building. The chapter will be closed with a justification regarding necessity of applying aggregation, in order to reduce the size of the modeled networks.

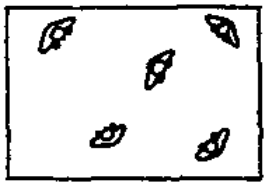
3.1 Modeling of Buildings in General

3.1.1 Nodes

Nodes can be used to model segments of a building such as rooms, hallways, lobbies or other locations. They may also represent an intersection point between two crossing walkways, where it is possible to change the direction. Think of a stadium, a theater or a lecture hall; here a node could also be used for modeling a single seat. If a room or a location has a large area, it is reasonable to divide the room or location into different sub segments. Each of these sub segments can be represented through a single node. By modeling the first floor of SAP's main building, we will have to apply such an approach for the Casino which has an area of 2700 m^2. It is advisable to model large rooms with more than one node, because we assume that a node, representing a particular location, is located in the middle of this location. This is an important point, considering the distance between neighboring locations or the distribution of evacuees. If we think of a room, for example, with an area of 1000 m^2 and model this room by only one single node, we would suggest that all occupants of the room are located in the middle of this room. Thinking of an ideal modeling each location a person can go to in a room should be modeled as a single node. However, this is not applicable in practice with the help of dynamic network flow models representing evacuation processes. As you can imagine so far, the accuracy of modeling has an effect on the results of the algorithms used to determine the evacuation time. These effects depending on the so-called "level of aggregation" will be discussed in one of the following chapters.
There are two main attributes of a node concerning the evacuation process: the initial number of occupants in the beginning of the evacuation and how many occupants can rest in the location represented through the node.
The *initial number of occupants* depends on the building type. In office buildings with restricted access only (e.g. for employees), it is easy to determine the initial number. Things get more complicated in buildings which have no access restriction such as job centers for example. The initial number of occupants for each location is represented through the so-called supply in the network model. A node representing a location which has a supply (i.e. initial occupants) is a source node. We have seen in the previous chapter that the source nodes are connected to a super source. It depends on the formulation of the problem if we connect all time copies to a super source or only the time copies for time unit zero (e.g. EAF vs. Quickest Flow). Possible emergency locations are represented through sinks. Because we do not stipulate which exit should be taken by the evacuees, we will connect all sink nodes to a super sink for all time periods. The super sink can be interpreted as a common safety area in the final model, which has a demand equal to the total supply (i.e. the total number of evacuees).

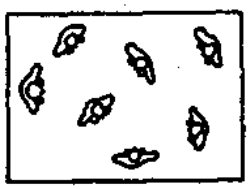
In order to calculate the *node (holdover) capacity* we need a formula which consists of two components. The *area occupancy factor* (AOF) and the *useable area* (UA) of a location. The area occupancy factor is the number of square meters allowed per person. The following table (taken from [KFN98] and defined by Fruin [Fru71]) shows different levels for the area occupancy factor and how these different levels can be interpreted.

QUEUING LEVEL OF SERVICE A



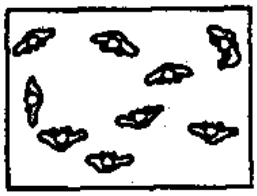
Average Pedestrian Area Occupancy: 1.21 sq.m./person or more
Average Inter-person Spacing: 0.37 m., or more
Description: standing and free circulation through the queuing area is possible without disturbing others within the queue.

QUEUING LEVEL OF SERVICE B



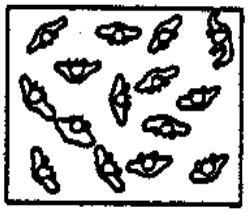
Average Pedestrian Area Occupancy: 0.93-1.21 sq.m./person
Average Inter-person Spacing: 0.33-0.37 m.
Description: standing and partially restricted circulation to avoid disturbing others within the queue is possible.

QUEUING LEVEL OF SERVICE C



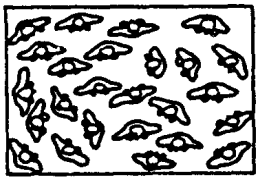
Average Pedestrian Area Occupancy: 0.65-0.93 sq.m./person
Average Inter-person Spacing: 0.28-0.33 m.
Description: standing restricted circulation through the queuing area by disturbing others within the queue is possible; this density is within the range of personal comfort.

QUEUING LEVEL OF SERVICE D



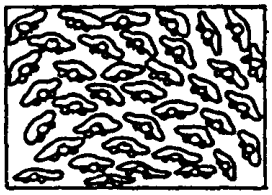
Average Pedestrian Area Occupancy: 0.28-0.65 sq.m./person
Average Inter-person Spacing: 0.19-0.28 m.
Description: standing without touching is possible; circulation is severely restricted within the queue and forward movement is only possible as a group; long term waiting at this density is discomforting.

QUEUING LEVEL OF SERVICE E



Average Pedestrian Area Occupancy: 0.19-0.28 sq.m./person
Average Inter-person Spacing: 0.19 m. or less
Description: standing in physical contact with others is unavoidable; circulation within the queue is not possible; queuing at this density can only be sustained for a short period without serious discom.

QUEUING LEVEL OF SERVICE F



Average Pedestrian Area Occupancy: 0,19 sq.m./person
Average Inter-person Spacing: close contact with persons
Description: virtually all persons within the queue are standing in direct physical contact with those surrounding them; this density is extremely discomforting; no movement is possible within the queue; the potential for panic exists in large crowds at this density.

Table I: Queuing Levels of Service defined by Fruin

If we take a *queuing level of service D*, for example, we get an area occupancy factor between 0.28 m^2 and 0.65 m^2 .This means that the number of square meters allowed per person is between 0.28 m^2 and 0.65 m^2.

The useable area is the second component required for calculating the node capacity. To get the useable area we subtract from the full area (width*length) the area reserved for obstructions like tables, wall closets and so on. We also consider a safety margin regarding the distance an occupant has to a wall. Therefore, we subtract 30 cm of each location's width and length. Finally, we get the following formula for the useable area of a location.

$$UA = (\text{width} - 0,3\,\text{m}) * (\text{lenght} - 0,3\,\text{m}) - \text{area reserved for obstructions}$$

We finally get the node capacity of node *i* by dividing the useable area by the area occupancy factor (AOF) and rounding to the nearest integer.

Node (Holdover) Capacity

$$h_i = \left[\!\left[\frac{UA_i}{AOF_i} \right]\!\right]$$

Remark: Floor loadings must be taken into consideration as well. This means that the total number of people that can rest at a location should not exceed the ration of the allowable floor loading to the average weight of an occupant.

It is also important to realize that other considerations may also be relevant when defining the node capacity. For example, if a node represents an office and no arcs entering the node, we may decide to set the node capacity equal the number of occupants of the office. A node capacity should always be at least as large as the initial content in order to be meaningful for the model.

The following algorithm summarizes the statements made so far concerning the representation of locations as nodes and how the node capacities can be determined.

INPUT: Blueprint of the building, segmentation into different locations which should be modeled as nodes

OUTPUT: Node representation for the different locations, with the corresponding node capacity

STEP 1: Assign to each location a node, located in the middle of the location

STEP 2: Compute the useable area (UA) of the location segment represented through the node

STEP 3: Assume an appropriate area occupancy factor (AOF) for the location segment

STEP 4: Compute the attentive node capacity, using the results of STEP 2 and STEP 3

STEP 5: If necessary modify the tentative node capacity in order to obtain the final node capacity by taking floor loading into consideration

STEP 6: Assign the supply, i.e. the initial number of occupants, to each node

Remark:

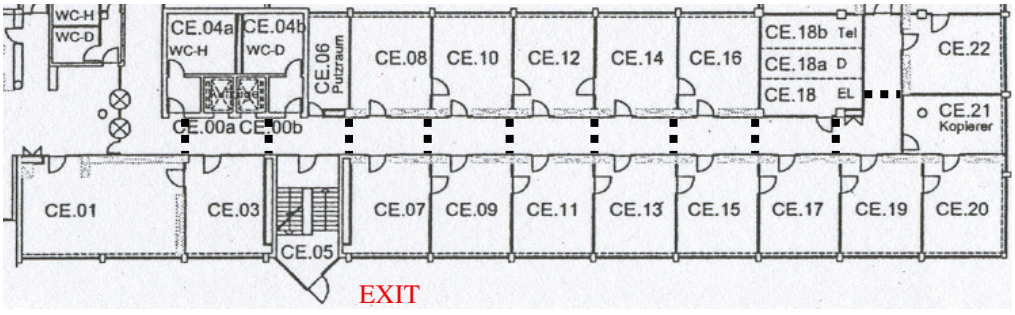
1.) The algorithm needs a blueprint which is segmented into different locations. How this segmentation should look like depends on the particular building.

2.) The connection of the sinks/sources to the super source/super sink is done in the final model.

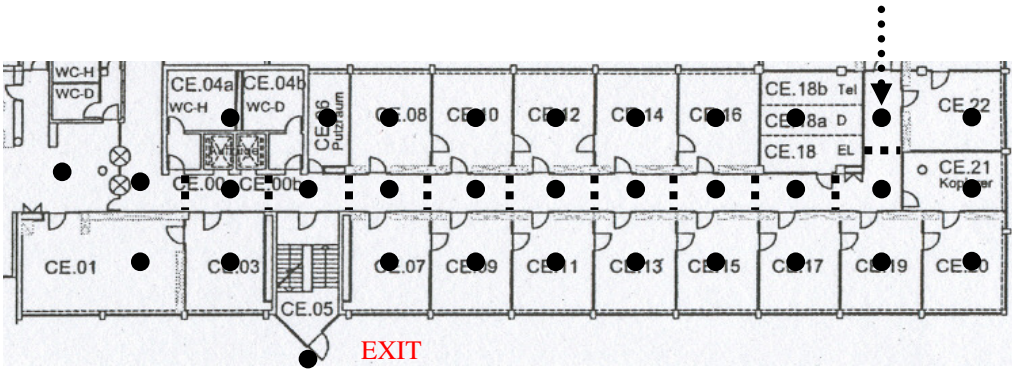
We take a part of the blueprint of the EVZ to visualize the proceeding of the algorithm in the following example.

Example 3.1

Below we see a possible segmentation of the EVZ's blueprint. Each room represents a location and the hallway is divided into different segments, represented through the dashed lines.



To each location a node is assigned



We have not modeled the staircase CE.05, because it will only be used for transition to the safety area.
For each node representing a room (e.g. an office), the capacity depends on the initial occupancy. For the segments of the hallway, the node capacity can be calculated as follows.

Assume we have an occupancy area factor of level D, we get the following range for the holdover capacity:

$$14 = \left\lfloor \frac{(2,2-0,3)*(5-0,3)}{0,65} \right\rfloor \leq h_i \leq \left\lfloor \frac{(2,2-0,3)*(5-0,3)}{0,28} \right\rfloor = 32$$

(We calculated the holdover capacity for the node marked with the dashed arrow in the figure above.)

Depending on the initial occupants in a location, the supply for each node has to be assigned.

3.1.2 Arcs

In general, arcs represent the connection between adjacent locations. For example, if a lobby is next to a hallway and it is possible to enter the lobby from the hallway, or vice versa, the connection can be represented by an arc. It depends on the given situation how the arc is directed. If we take, for example, office CE.03 of the blueprint in Example 3.1, it is not advisable to have an arc directed into the office. One arc leaving the node, representing office CE.03, seems to be enough, since it is the only reasonable direction an evacuee can take. In contrast, arcs connecting hallway segments should be directed into both directions, because an evacuee can often choose between more than one evacuation route. In our model, an arc indicates if two locations are connected or not. However, to get a full description of the arc, we also need information about the characteristics of the connection. For our case it is sufficient to know how long it takes to travel from location *i* to location *j* and, additionally, how many people can travel at once per time unit on this connection.
It does not seem to be a crucial part to compute the travel time between two locations. Dividing the distance between both locations by the travel speed would yield the travel time. The distance between two locations should be estimated to be the median distance that people require to travel from one location to another. Therefore, we take the center of both locations connected through the arc and calculate their distance. Here we have to take the specific building characteristics into account. The following figure shows such a situation. We have to take the distance represented through the solid line instead of taking the shortest distance represented through the dashed line.

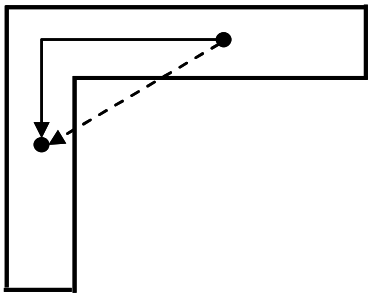
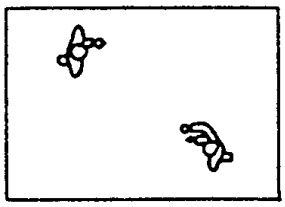


Figure 5: Distance between two locations

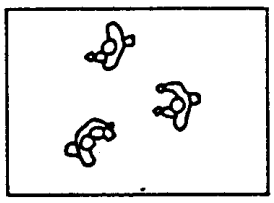
We have ignored two important factors computing travel times so far: besides human factors (e.g. age, gender etc.) the travel time depends also on the flow density. In our approach, different from the microscopic approach, we assume a homogenous group of evacuees. Therefore, the human factor can be neglected. However, it is not possible to neglect the density. If the density is high on a connection, individuals have to conform their speed to the speed of the mass and congestion may occur. Although we take account of this fact, our model maps only a part of the reality. We use once more the level of service concept, defined by Fruin. Depending on the area occupancy factor, he defined besides different *average flow volumes* in persons per meter per minute also different *average speed values*. Using these parameters we are able to calculate travel times and capacities which depend on the *crowd level*. But as mentioned above, our model maps only a part of the reality, because we assume a constant crowd level for the whole evacuation process. If we want to have a more realistic modeling we would have to use flow\density dependent travel times, as it is done in traffic assignment for example (e.g. [CS00],[JTC95]). The different level of services (LoS), taken from Fruin, are shown in the following table.

WALKWAY LEVEL OF SERVICE A



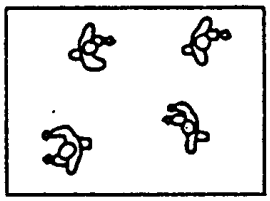
Average Flow Volume: 23 PMM* or less
Average Speed: 79 m./min.
Average Pedestrian Area Occupancy: 3.26 sq.m./person or greater
Description: Virtually unrestricted choice of speed; minimum maneuvering to pass; crossing and reverse movements are unrestricted; flow is approximately 25% of maximum capacity.

WALKWAY LEVEL OF SERVICE B



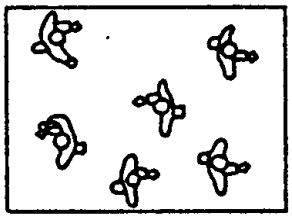
Average Flow Volume: 23-33 PMM*
Average Speed: 76-79 m./min.
Average Pedestrian Area Occupancy: 2.33-3.26 sq.m./person or greater
Description: normal walking speeds only occasionally restricted; some occasional interference in passing; crossing and reverse movements are possible with occasional conflict; flow is approximately 35% of maximum capacity.

WALKWAY LEVEL OF SERVICE C



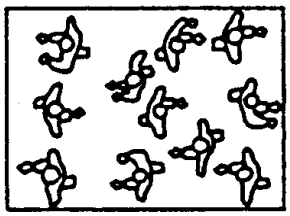
Average Flow Volume: 33-49 PMM*
Average Speed: 70-76 m /min.
Average Pedestrian Area Occupancy: 1.4-2.33 sq.m./person or greater
Description: walking speeds are partially restricted; passing is restricted but possible with maneuvering to avoid conflict; flow is reasonably fluid and is about 40-65% of maximum capacity

WALKWAY LEVEL OF SERVICE D



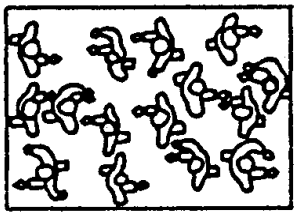
Average Flow Volume: 49-66 PMM*
Average Speed: 61-70 m./min.
Average Pedestrian Area Occupancy: 0.93-1.4 sq.m ./person
Description: walking speeds are restricted and reduced, passing is rarely possible without conflict; crossing and reverse movements are severely restricted with multiple conflicts; some probability of momentary flow stoppages when critical densities might be intermittently reached; flow is approximately 65-805 of maximum capacity.

WALKWAY LEVEL OF SERVICE E



Average Flow Volume: 66-82 PMM*
Average Speed: 34-61 m./min.
Average Pedestrian Area Occupancy: 0.47-0.93 sq.m./person
Description: walking speeds are restricted and frequently reduced to shuffling; frequent adjustment of gait required; passing is impossible without conflict; crossing and reverse movements are severely restricted with unavoidable conflicts; flows attain maximum capacity under pressure, but with frequent stoppages and interruptions of flow.

WALKWAY LEVEL OF SERVICE F



Average Flow Volume: 82 PPM or more*
Average Speed: 0-34 m./min.
Average Pedestrian Area Occupancy: 0.47 sq.m./person or less
Description: walking speed is reduced to shuffling; passing is impossible; crossing and reverse movements are impossible; physical contact is frequent and unavoidable; flow is sporadic and on the verge of complete breakdown and stoppage.

Table II: Walkway Levels of Service defined by Fruin (taken from [KFN98])
*PMM = Persons per meter width of walkway, per minute.

Once an area occupancy factor respective a level of service has been chosen, we can compute the travel time between two locations *i* and *j* by using the following formula:

Travel Time

$$\lambda_{ij} = \left[\!\!\left[\frac{\text{distance from location } i \text{ to } j \text{ in meter}}{\text{average speed in meter per second} * \pi} \right]\!\!\right]$$

As we have seen in Chapter 2, the time horizon T of a dynamic network is broken up into uniform, discrete time units $t = 0,1,\ldots,T$. π is the basic time unit, defining the length of one time period. Hence, the measuring of travel times also depends on π. For example, if the length of one time unit is three seconds, e.g. π=3, then time $t = 4$ is associated with a real time of 12 seconds. For the original evacuation problem, we assume that π is equal to one. This means that travel times are measured in seconds.

If a level of service has been chosen, it is also no difficulty to compute the capacity of an arc. The multiplication of the *average flow volume* with the minimum *useable width* leads to the capacity of the arc. The term "useable width" means that we subtract 0.3 m from the real width, since no one would rub along a wall, even in the case of emergency. It is also important to take the minimum width of the path, because the capacity depends on the section with the lowest width. In most cases, this will be a doorway of some sort between the two locations; see the following figure.

Figure 6: Capacity restriction of an arc connecting two locations

Even though both locations have the same width, we would take the width of the doorway computing the capacity of the arc, since the doorway can be seen as a "bottleneck" for this connection.

Finally, the capacity per time unit of an arc representing the connection between two locations *i* and *j* can be calculated as follows:

Arc Capacity

$$u_{ij} = \left[\!\left[(\text{minimum useable width} - 0.3\text{m}) * \left(\frac{\text{Average Flow Volume in PMM}}{60\,\text{sec}} \right) * \pi \right]\!\right]$$

As in the case of computing the node capacity it should be clear that the flow capacity is an upper bound on the actual flow and may not actually be achieved.

The following algorithm summarizes the different steps, computing the capacity and travel time of arcs connecting two locations.

INPUT: Blueprint of the building, segmentation into different location which are already represented as nodes

OUTPUT: Arcs connecting nodes, which represent adjacent locations; network representation of the blueprint

STEP 1: Choose an appropriate level of service

STEP 2: If a location represented by a node i is adjacent to a location represented by a node j and j can directly be reached from i (e.g. by a door), then add an arc (i, j)

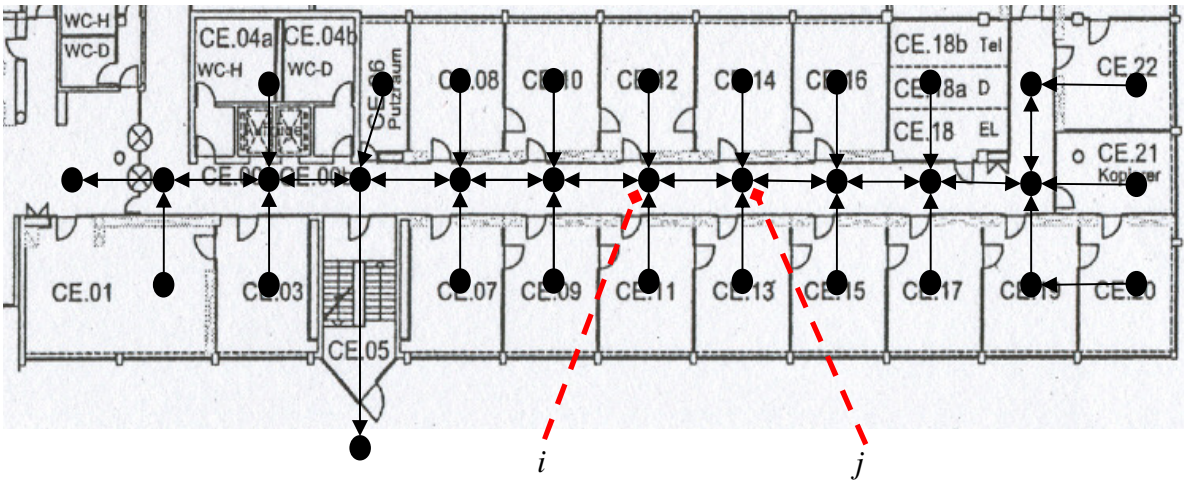
STEP 3: For each arc (i, j), representing the connection of two locations, multiply the minimum useable width with the average flow volume in seconds, multiply the result with the basic time unit π and round the result to the nearest integer. The result will be the capacity of the arc.

STEP 4: For each arc (i, j), divide the distance between the corresponding locations by the basic time unit π multiplied with the average speed per second corresponding to the chosen level of service. The result will be the travel time of the arc

Remark: In STEP 2 it is up to the user to define the term adjacent.

This section concludes with the following example, in which we assume a basic time unit π=3 and a level of service D.

Example 3.2



Computing the travel time and the capacity for arc (i, j) yields the following results:

$$1 = \left\lfloor \frac{4.8 * 60}{70 * 3} \right\rfloor \leq \lambda_{ij} \leq \left\lfloor \frac{4.8 * 60}{61 * 3} \right\rfloor = 2$$

i.e. the travel time is between 1 and 2 time units.

$$5 = \left\lfloor (2.4 - 0.3) * \frac{49}{60} * 3 \right\rfloor \leq u_{ij} \leq \left\lfloor (2.4 - 0.3) * \frac{66}{60} * 3 \right\rfloor = 7$$

i.e. the capacity is between 5 and 7 persons per time unit.

Remark: For the complete model we have to add a super source and a super sink.

3.2 Case Study: Modeling one floor of SAP's Main Building

In the following, we will apply the theory regarding the modeling of evacuations objects to our real world problem. We have mentioned before, that our aim is the computation of the clearing time for the ground floor of SAP's main building, the EVZ. First we have to derive the network representation of the building in order to use algorithms for such a computation. Since the modeling mainly depends on the particular building characteristics, we must have a closer look on the blueprint. The ground floor can be separated into three segments, representing different architectural patterns.

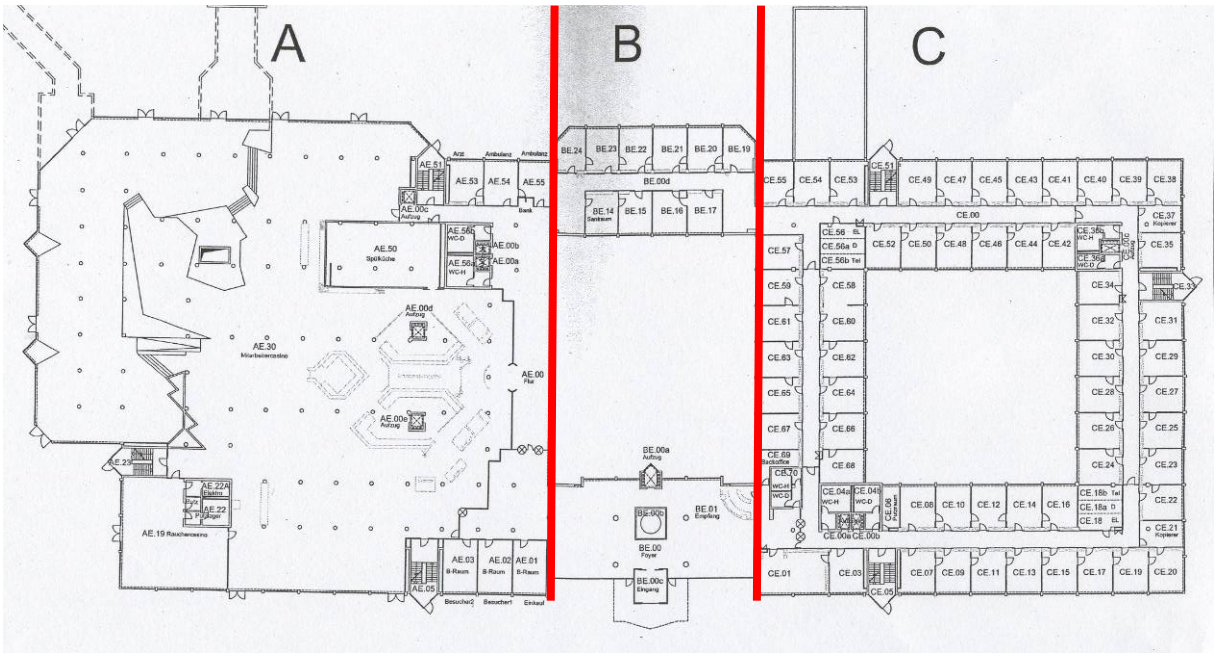


Figure 7: Segmentation of the EVZ

The first pattern which can be identified is Sector A, the Casino. The Casino has a large and unstructured area of nearly 2.700 m^2. The term "unstructured" means that at a first view it is very difficult to divide the Casino into reasonable sub segments.
The Casino has an overall capacity for about 3000 people. During lunch time, i.e. from half past eleven to 2 o'clock, we can observe about 900 persons, taking their meal there. Because the employees use the Casino as a coffee zone, the Casino is not closed and empty before the lunch time. Besides the main usage as a canteen, the Casino is also used for employee meetings or the companies Christmas Party, where it is possible that the maximum capacity of the Casino is reached. For our evaluation, we assume that about 700 persons are in the Casino, when the evacuation takes place.

The second architectural pattern which can be identified is Sector C, the Office Complex. Even though this sector has nearly the same area as the Casino (2.500 m^2), it is strictly divided into different logical and physical sub segments represented by the different rooms. Most of the rooms are used as offices. Besides offices there are also restrooms and storage rooms for technical and cleaning stuff.

Since the large area is divided into sub segments in advance, the modeling of such areas will be much easier than for the Casino. The different offices are in most cases assigned to particular employees. Therefore, it is possible to determine the initial number of occupants. We assume an initial occupancy of three persons per room.

The third and last architectural pattern is Sector B, the Buffer Zone. The Buffer Zone mainly represents the connection between Sector A and Sector C and consists of the large lobby (BE.00) and the hallway (BE.00 d), with some offices and the ambulance.

Sector B is some kind of special characteristic of the EVZ, whereas the segments represented by the Casino and the Office Complex can be found in nearly all company buildings.

Since we can identify three different segments, it will be reasonable to take this segmentation for the modeling, too. For the modeling we can omit Sector B, because it is some kind of mixture of the components of Sector A and Sector C and is mainly used as a passage way. Therefore it is enough to represent it by two single nodes and corresponding arcs in the final model.
In the following, we will model the Office Complex as well as the Casino in detail. By doing so, we discuss the different problems coming from the modeling of large rooms in general and for the Casino in particular.

3.2.1 Modeling the Office Complex

As in the last sections, locations are modeled as nodes. Concerning the Office Complex, we can identify three different kinds of locations. The first and main share of locations is represented by rooms. Most of the rooms are used as offices. The remaining ones are rest rooms (e.g. CE.04) and rooms for storing technical or cleaning equipment. As we have mentioned above we assume an initial occupancy of three persons per room. Each room will be represented by a node. Because of the fact that the different rooms have an initial number of occupants, the nodes representing the rooms in the final dynamic network will be sources. The capacity of nodes representing rooms is equal to the initial occupancy of the rooms. In the final model the sources are connected to a super source.

The second kind of locations which can be identified are the safety areas. They can be reached by using the emergency exits of the staircases CE.05, CE.33 and CE.51. It is also possible to leave Sector C by using hallway BE.00d or by using the sally port which directs to the lobby. In terms of evacuation, most of the occupants in the Office Complex would take emergency exits CE.05, CE.33, CE.51 or the sally port. The safety areas are represented through sinks in the network model. All the sinks are connected to a super sink which can be interpreted as a common safety area and has a demand which is equal to the total supply (i.e. the total number of occupants).

The last kind of locations which can be identified are the different hallway segments. We decided to divide the complete hallway into logical segments, in order to have a more realistic model. Two opposite rooms share one hallway segment whenever possible. Since we assume that there are no evacuees on the hallway in the beginning of the evacuation, the hallway segments are represented through transshipment nodes (i.e. nodes which have neither supply nor demand). The capacity of the different segments can be calculated by using the formula of the last section. For the calculation we assume an area occupancy factor of *0.5 m^2/person*. The

hallway segments could be seen as an entry point into the evacuation route system of the Office Complex.
In the following figure it is shown how the segmentation of the hallway into different segments has been done.
The dashed lines show the segmentation of the hallway into different sub segments.

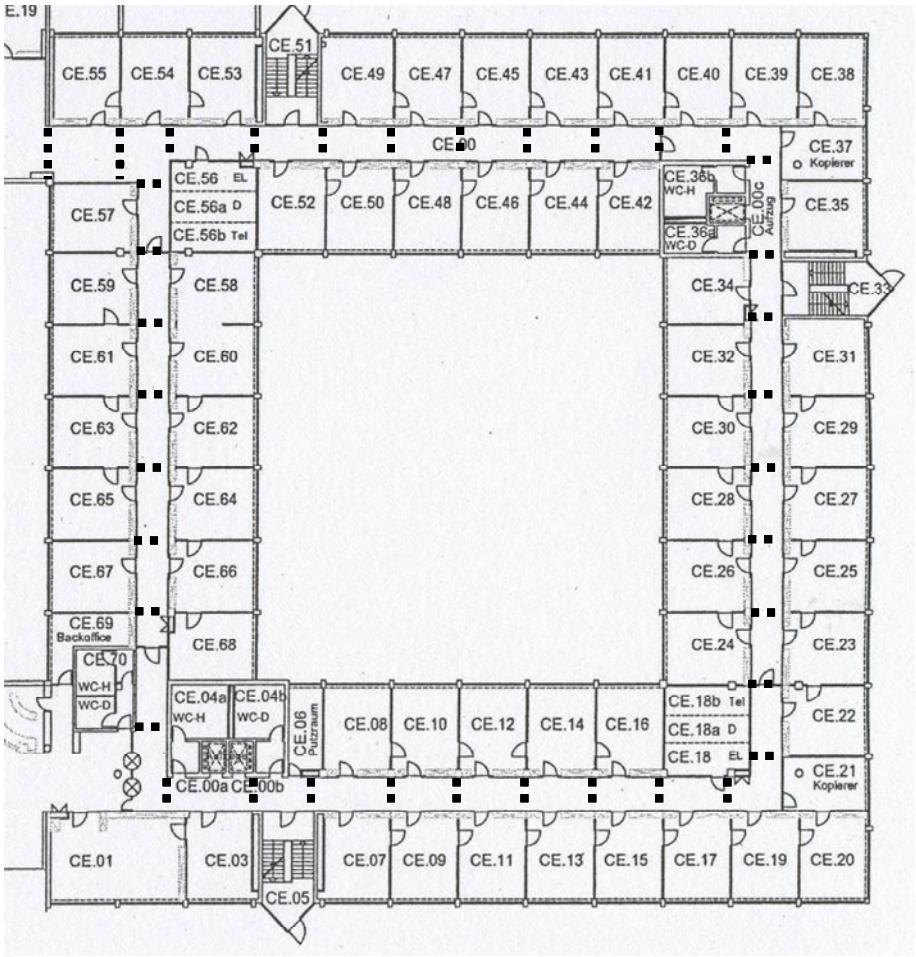


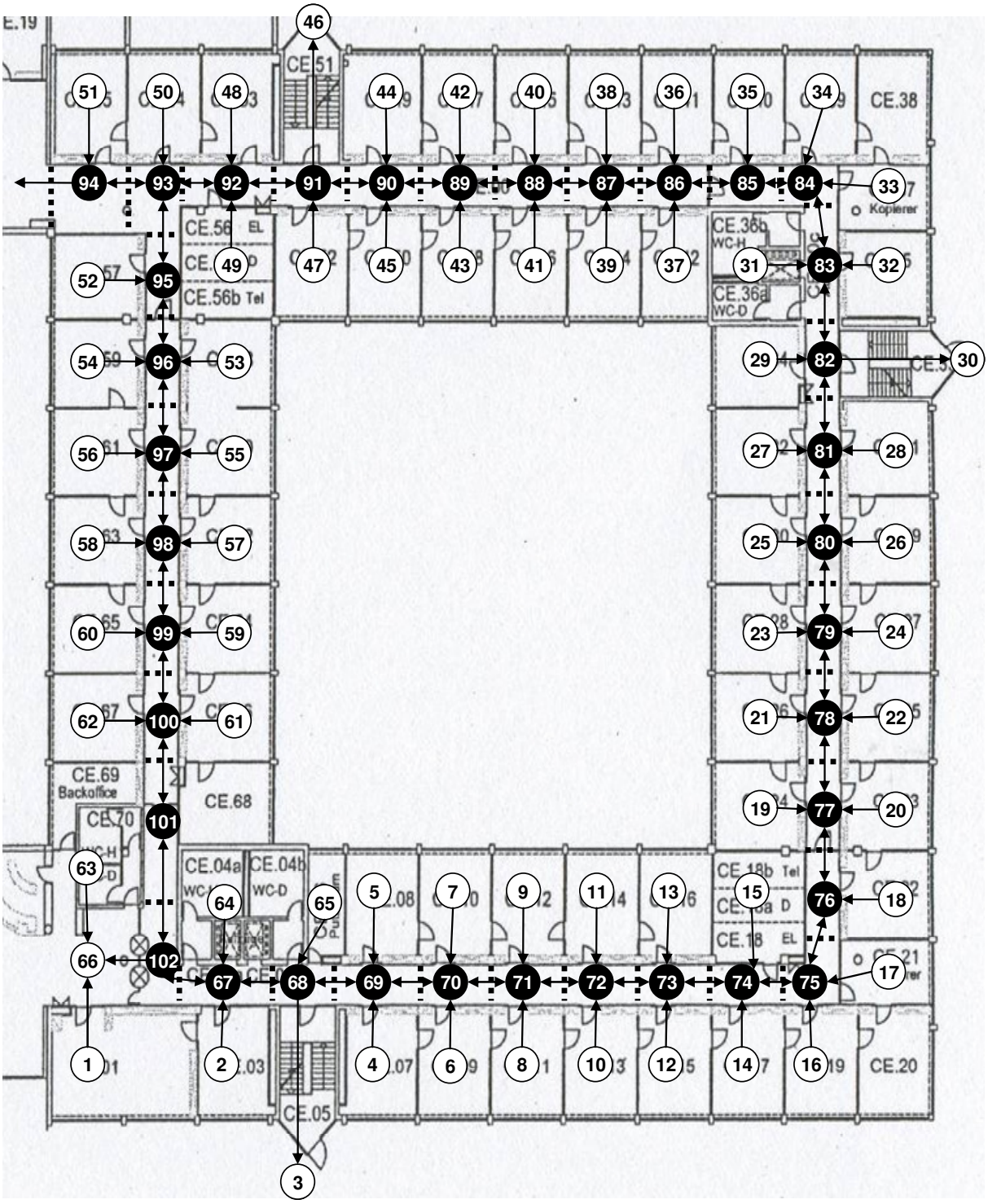
Figure 8: Segmentation of the hallway of the Office Complex

So let us have a look at the representation of the different connections between the locations.
For each arc connecting two locations, we assume a Level of Service D for the calculation of the travel time and capacity.
Each room is connected to its particular hallway segment by a directed arc, where we assume that only one arc is leaving the room and no arc entering the room exists. For most arcs leaving rooms, we finally get a travel time of *4 seconds* and a capacity of *1 person per second*. Adjacent hallway segments are also connected by arcs, which are directed in both directions. Based on the LoS D, the capacity and travel time can be calculated as shown in the last section. To get from the hallway segments in front of CE.05, CE.33 and CE.51 to the corresponding emergency exit, we have an arc leaving the hallway segment in direction to the safety area. We get a travel time of *9 seconds* and a capacity of *1 person per second* for passing one of the emergency exits located in the staircases CE.05, CE.33 and CE.51. Whereas we have a travel time of *15 seconds* to pass the sally port and get to the lobby. To get to the hallway BE.00d, the general computations for the travel time and capacity can be used. We will conclude this section with the presentation of the complete model for Sector C in terms of a dynamic network. We set the basic time unit π equal 1 (e.g. the travel times are

measured in seconds). The parameters corresponding to the example can be found on the enclosed CD-ROM.

Example 3.3

Since the rooms CE.20, CE.38 and CE.68 have no direct access to the hallway we can omit them.



Note: Due to visualization we omit the artificial nodes, which are needed to represent arcs directed into both directions. We also omit the super sink and super source.

3.2.2 Modeling the Casino

Before modeling the Casino in detail, we will first discuss the problems coming from the modeling of large rooms in general. In our definition, large rooms are rooms with an area larger than 100 m^2. At first glance, it is often not possible to identify physical structures in large rooms, which separate the room into different sub segments. However, an adequate modeling requires the identification of such sub segments, since it is not reasonable to model a room such as a Casino or a Concert Hall by using only one node. If we take only a single node for representing a large room, we would suggest that all the occupants stay in the middle of the room. For small rooms this is not a problem, but for large rooms this would lead to a great loss of details. Therefore, we have to think about possibilities to divide large rooms into reasonable sub segments.
If we take the modeling of the Office complex in the last section for instance, we could observe that even though the overall area was about 2.500 m^2, it was possible to divide the area into logical, physical and recurring patterns. This means that the modeling was somehow streamlined. The division into different sub segments was given in advance and we did not have a wide range of alternatives for the modeling, except the modeling of the hallway. Each room, exit and hallway segment was represented through a node. It was also possible to model all the evacuation routes that a single evacuee could take because of the fact that all the routes were predetermined through the hallway. Therefore, it was sufficient to model besides the different hallway segments and their connections, a connection from the offices to the particular hallway segments. By comparing the characteristics coming upon in the case of the Office Complex, the situation for large rooms is more complicated. It is neither given nor obvious how to divide the room into different sub segments and what degree of detail should be taken concerning the modeling. It is also not trivial to determine which evacuation routes should be mapped, because there are no physical hallways which streamline the flow of evacuees. As we can see in the following figure, we have to distinguish between two kinds of evacuation routes. Evacuation routes which have the shortest distance to the safety area for a particular occupant on the one hand and the consolidated standard evacuation routes which are predetermined by us on the other hand.

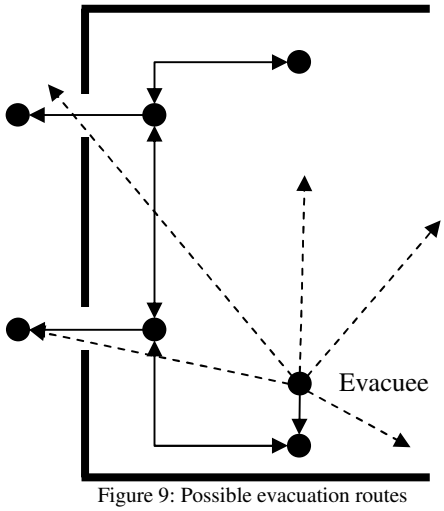


Figure 9: Possible evacuation routes

The solid arcs in the figure represent the standard or mapped evacuation route whereas the dashed line represents some of the possible routes for the particular evacuee. Therefore, we have to accept a tradeoff between the individual and the standard routes, because it makes no sense to model each individual route.

As mentioned above, it makes no sense to model a large room as a single node. Therefore, we derived an approach which divides a large area into virtual, separated sub segments that have a standard size. We use the concept of the so-called "virtual grids". Each virtual grid element is represented through a node located in the middle of the element. The area represented through the node is defined by the size of the grid (i.e. a node corresponding to a grid of 5 m width and 5 m length represents an area of 25 m^2)
The virtual grid element could be interpreted as a "virtual room" which can be left in all directions. The following figure shows a virtual grid element representing an area of 25 m^2.

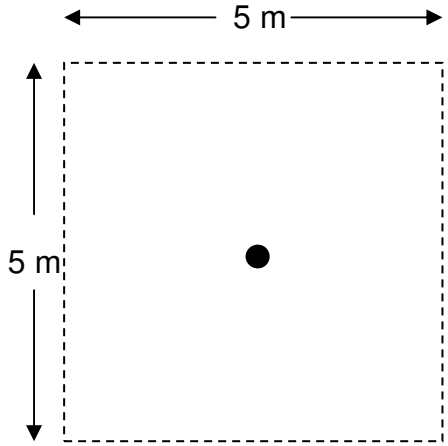
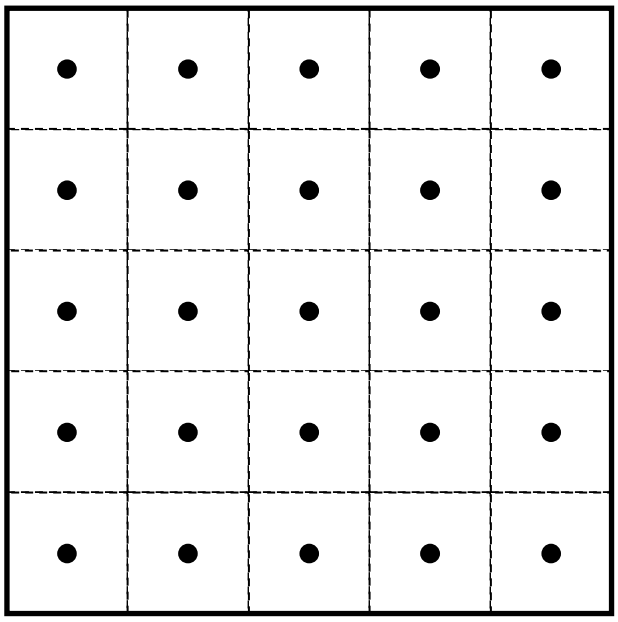


Figure 10: A virtual grid element

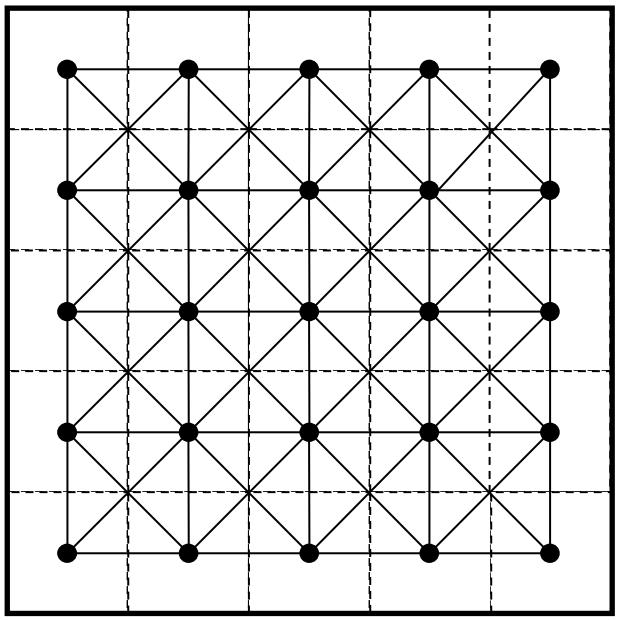
If we place the virtual grid elements side by side and one below the other, we get the complete virtual grid, which corresponds to a segmentation of the former large room. To get an impression on how the approach leads to a segmentation of (large) rooms, the following example can be useful.

Example 3.4

Assume we have an area of 50 m width by 50 m length and a size of 10 m width and 10 m length for each grid element. Then we would finally get the following sub segments for the room.



If we connect the different nodes through edges, we would finally get the following network representation for the large room.



Note: This approach does not consider the particular room characteristic of the room, i.e. there might be parts of the room represented through a node which can not be passed.

We can set the size of each virtual grid element equal to the area occupancy factor in order to include more details of the problem characteristics and allow an individual route selection. If we take the AOF as the size of each virtual grid element we would finally have the same basic approach used for cellular automat (see for example [BA99], [KMSW00]).
It is obvious that the more details we want to model the more effort and data we get. If we have to model large rooms (e.g. concert halls) in which it is not possible to identify reasonable sub segments we should use *microscopic models* like cellular automates rather than the optimization approaches based on dynamic networks. If it is possible to identify structures which can be taken for defining reasonable sub segments, they should be adopted in the model. We can also adapt our approach by allowing virtual grid elements of different size corresponding to the problem specific characteristics. Such an adoption will also be made for modeling the Casino.
When we talk about the modeling of the Casino we must have a closer look on the blueprint and try to identify architectural characteristics of the Casino. It is possible to use some of them for a reasonable modeling of the Casino.

- Most of the area is reserved by different groups of tables
- Further parts of the area is used by the food-station
- A fountain and some flowerbeds require also a part of the Casino's area
- Due to the positioning of the groups of tables and the food-station the possible emergency routes are restricted

- We can further identify 14 emergency exits which can be reached directly from the Casino

- The lobby can be reached through three sally ports

In the following figure the identified characteristics are marked on the blueprint

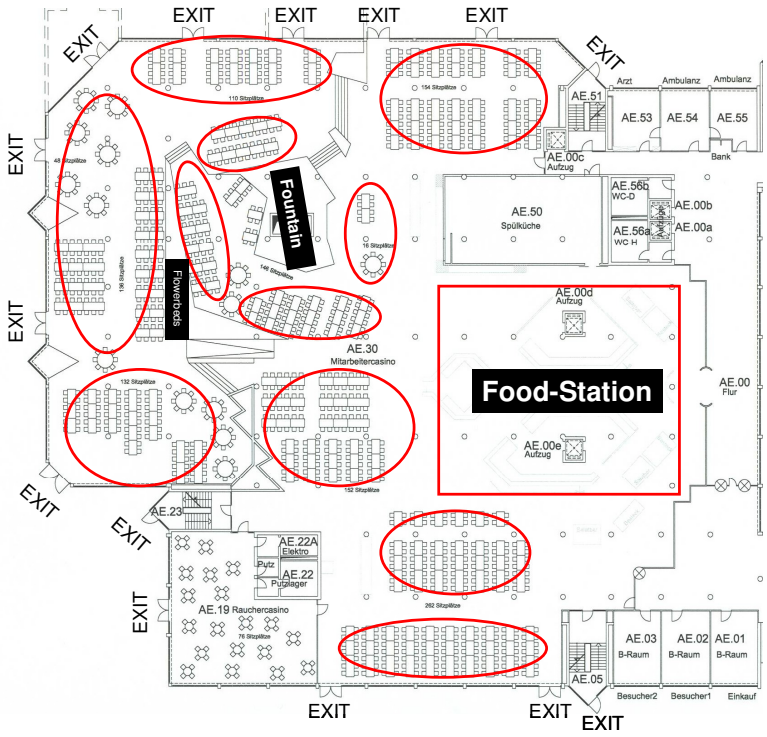


Figure 11: Highlighted characteristics of the Casino

We have to find a compromise between model accuracy and modeling effort. The observations made so far encourage us to apply the perceptions made for the modeling of the Office Complex also for the Casino. Therefore, we treat the area used by the tables as *virtual rooms*. They can be left at most into two directions. As it was also the case for the Office Complex the virtual rooms are modeled as nodes. The capacity of such nodes depends on the initial number of occupants.
We also define *virtual hallways*, which are predetermined by the location of the different groups of tables, the fountain and the food-station. As it was done for the Office Complex, we divide the virtual hallway into different segments corresponding to the different virtual rooms. By taking a Queuing Level of Service D, we can calculate the node (holdover) capacity of the hallway segments. The following figure shows the virtual rooms and virtual hallways as well as the corresponding virtual hallway segments which can be identified. The segmentation can be also interpreted as a virtual grid, which is customized for the particular problem instance.

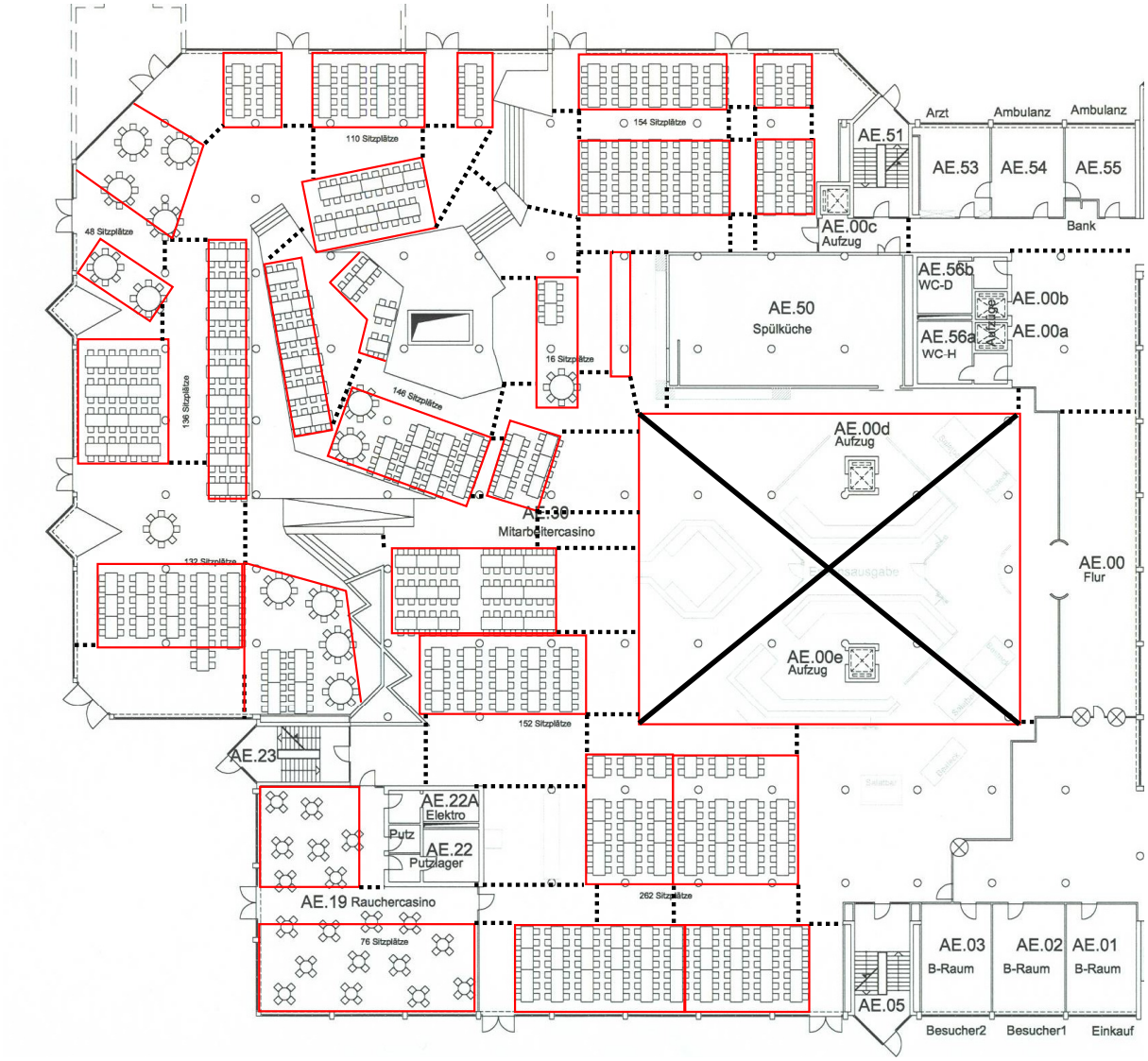


Figure 12: Casino segmented into different virtual rooms and hallway

Each node representing a group of tables has at most two arcs leaving the node in the direction of a particular hallway segment. There are no arcs entering such nodes, since we assume that no one will walk or jump over tables. As in the case of the Office Complex, adjacent hallway segments are connected through an arc directed towards both directions. We assume a LoS D for computing the corresponding capacities and travel times. We further assume a capacity of two persons per second for arcs passing one of the emergency exits which are not located in the staircase. Arcs passing one of the sally ports have a capacity of one person per second. So let us have a look on the final representation of the Casino. The corresponding parameters can be found on the enclosed CD-ROM.

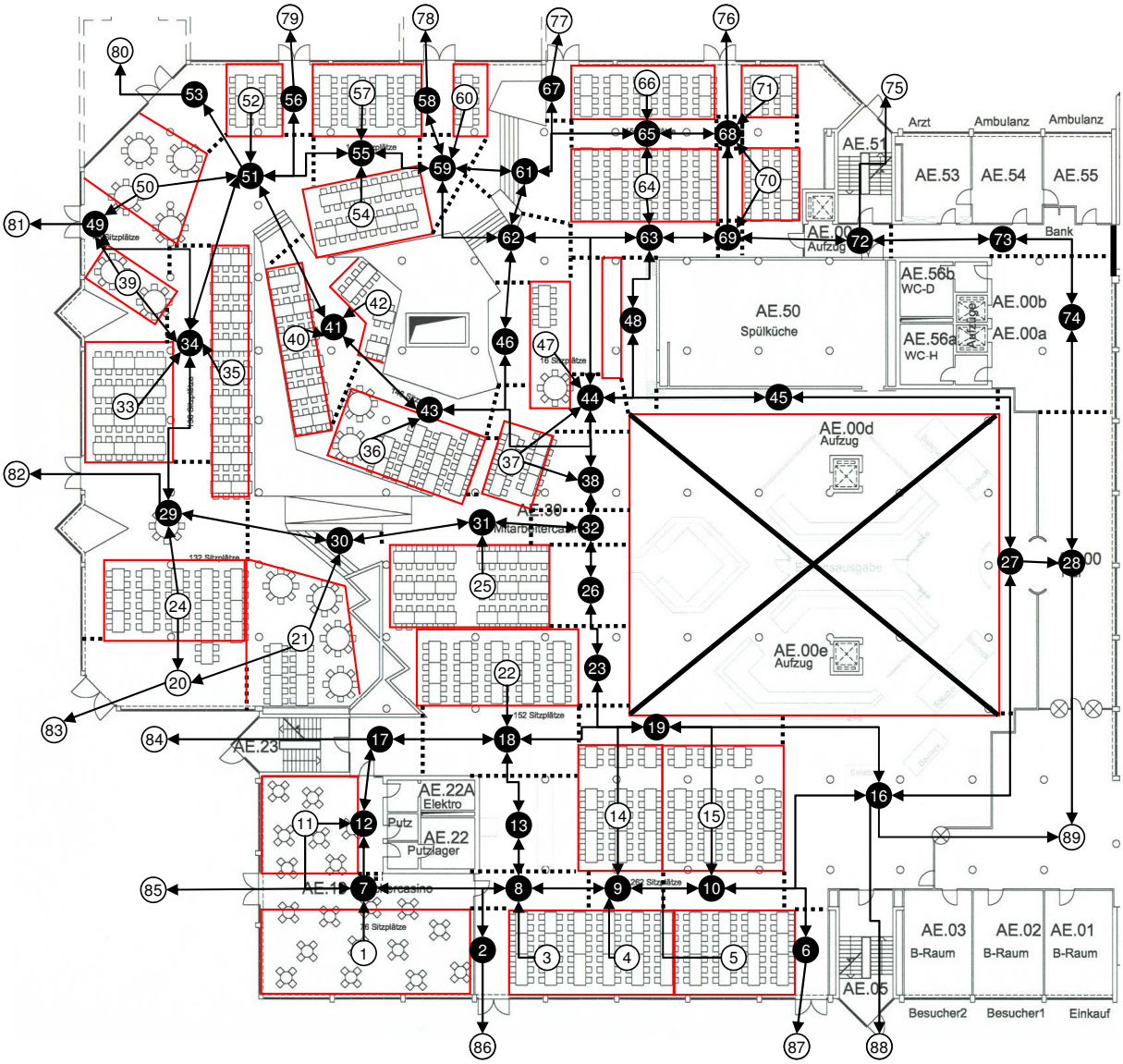


Figure 13: Final representation of the Casino as a network

3.3 The Need for Aggregation

If we combine the two network models of the Casino and the Office Complex and further include the Buffer Zone, we get the following network representation.

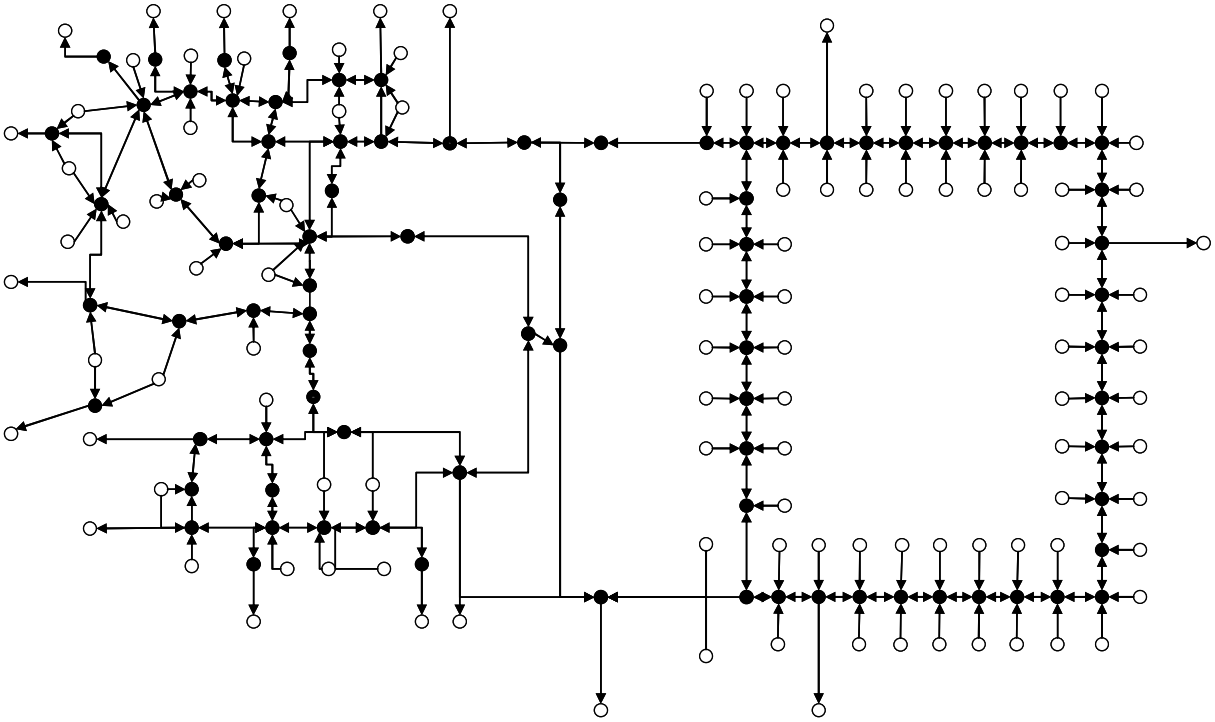


Figure 14: Complete network representation of the EVZ's ground floor

The network representation of EVZ's ground floor consists of 194 nodes and 210 arcs. We have to use a dynamic network to model the evacuation process. If we assume a time horizon of 100 seconds and a basic time unit $\pi = 1$, we get approximately 20.000 nodes and 40.000 arcs for the time expanded representation. We only modeled the ground floor of the EVZ, which has about 5 floors in total. This means that modeling the whole building would result approximately in 100.000 nodes and 200.000 arcs in the time expanded network. Because the algorithm we will use for calculating the evacuation time operates on the time expanded network we have to think about strategies to handle this large amount of data. We can think of two opportunities:

- Well designed algorithms for large network flow problems
- Reducing the degree of detail regarding the modeling

The latter opportunity results in less nodes and arcs, whereas the first opportunity would still require the complete data but would only use parts of it.
It seems to be more reasonable to reduce the modeled details by making a so-called *aggregation*. However, this would lead to a loss of accuracy regarding the final result (e.g. the evacuation time). In the following we will investigate the concepts of aggregation for transportation problems and minimum cost network flow problems in general and examine the effects of aggregation on the optimal result in particular. We will also review an algorithm for large network flow problems, which uses the concepts of aggregation, before we try to apply the results found in literature to the evacuation problem.

Chapter 4

Aggregation of the Transportation Problem

Let us start this chapter with a very interesting real world application which can be formulated as a transportation problem. Let us assume we are in a city which has a number of districts. After a snowfall, the snow in each area must be moved out of the district into a convenient location. These locations could be for example large grates leading to the sewer system, large pits or entry points to the river. Each of these destinations has a capacity. The goal is to minimize the distance traveled to handle all of the snow. This problem can be formulated as a transportation problem. In such a problem, there is a set of nodes called *sources*, and another set of nodes called *destinations*. All arcs are directed from the sources to the destinations. There is a per-unit cost on each arc. Each source has some kind of supply and each destination has a demand. We assume that the total supply equals the total demand (adding a fake source or destination if needed). If we take the snow removal problem, sources are equal to the locations where snow has to be removed. Destinations (sinks) are locations where the snow can moved to. As cost of the arcs we can think of the distance between the districts and the convenient locations. Our aim is to remove the snow where the mileage should be kept as small as possible.
Among other applications transportation problems are often used in transportation planning. We can think of a company which owns *m* warehouses and *n* retail shops. A single product has to be shipped from the warehouse to the shops. The warehouses represented through the sources store a particular amount of the product, whereas the retail shops represented through the sinks have a particular demand on the product. On each connection, transportation costs are defined, giving information about the costs (dependent of the distance, carrier-possibilities, etc.) of shipping one unit of the product from a warehouse *s* to a retail shop *d.* The problem of interest is to determine an optimal transportation plan between the warehouses and the outlets, subject to the available supply and demand. Optimality under this setting means that the total transportation costs should be as low as possible.
The transportation problem has a couple of useful properties. As long as the total supply is equal to the total demand a feasible solution exists. All the coefficients are equal to one and every flow variable x_{sd} appears exactly in two constraints. For most real world applications, it is also important that a solution of the transportation problem is integral. It is not necessary to have a constraint ensuring this integrality, because of the fact that as long as a feasible solution exists and the demand as well as the supply of each source respective destination is integral, an optimal integral solution exists.
It is possible to use the simplex algorithm to get optimal solutions. Due to the specific structure of the transportation problem, special methods such as the *u-v method* can be applied, which are more suitable for solving such problems.
Even with the growing capability of IT-Systems, the question of how to handle large transportation problems is still interesting. A possibility to handle large transportation problems is aggregation. Since the early sixties a lot of work has been done on evaluating aggregation for transportation problems. Much of the concepts were derived from results

gotten for the aggregation of general linear problems. See for example the work of Lee [Lee75]. Almost all approaches concerning the aggregation of transportation problems start with an unaggregated transportation problem (UATP), derive an aggregated transportation problem (ATP), solve the ATP and disaggregate the solution of the ATP into a solution for the UATP. A solution derived in such a way can be used as a fairly good initial solution for the original problem, as Balas [Bal65] has done it. The algorithm of Balas leads to an optimal solution for the original problem at the end. If we are not interested in an optimal solution for the original problem and take instead the approximate solution derived by the aggregated problem it would be interesting to know how good our solution is. Therefore, authors such as Zipkin [Zip80] or Taylor [Tay83] derived bounds on the error which is made by solving the aggregated problem instead of the original, unaggregated problem. This means that after solving the ATP and disaggregating the solution you can decide whether or not the solution provided by solving the ATP is good enough for the particular problem.
In the following section we will start with a problem description and definitions concerning aggregation which are valid for the approaches of Balas and Zipkin. In Section 4.2 we discuss the concepts of Balas, which can be named as the *aggregation by dominance*. He focused on an algorithm, which finally leads to an optimal solution for large-scale transportation problems. In Section 4.3 we continue with a section about the approach of Zipkin, the *weighted aggregation.* His approach finally leads to a feasible solution for the original problem. He also derived bounds on the quality of this solution. The chapter will be concluded with a discussion which of the both presented approaches is preferable.

4.1 Problem Description

Before we discuss the concepts of aggregation and disaggregation regarding transportation problems it will be necessary to have a formal description of the problem.

Unaggregated Transportation Problem

(UATP)

$$\min \sum_{s\in S, d\in D} x_{sd} c_{sd}$$

s.t.

$$\sum_{d\in D} x_{sd} = a_s \qquad \forall s \in S$$

$$\sum_{s\in S} x_{sd} = b_d \qquad \forall d \in D$$

$$x_{sd} \geq 0 \qquad \forall s \in S, d \in D$$

Where:

x_{sd} = flow from source $s \in S$ to destination $d \in D$

c_{sd} = cost for shipping one unit of flow from source $s \in S$ to destination $d \in D$

$S = \{1,...,n\}$ the set of sources

$D = \{1,...,m\}$ the set of destinations

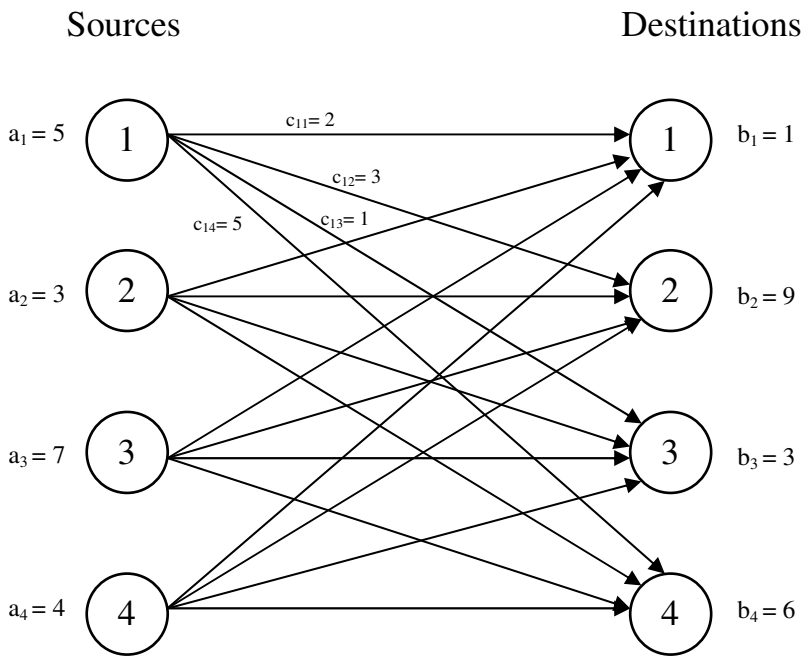
a_s = the (positive) supply of source node $s \in S$

b_d = the (positive) demand of destination node $d \in D$

Let us denote with $x^* = (x_{11}^*, ..., x_{sd}^*, ..., x_{nm}^*)$ any optimal solution for the UATP, with the corresponding objective value $z^* = \sum_{s \in S, d \in D} x_{sd}^* c_{sd}$. Further we assume in the following that $\sum_{s \in S} a_s = \sum_{d \in D} b_d$ (i.e. a feasible solution exists).

The following example provides an overview of the definitions already made and serves us as a basis for the examples coming up in this section.

Example 4.1



Sour. / Dest.	1	2	3	4
1	2	3	1	5
2	9	2	4	8
3	2	2	1	3
4	7	8	2	1

Table III: Costs of Example 4.1

Applying aggregation to an original transportation problem means that we group nodes together. Therefore, every aggregation will be based on a partition $\overline{SP}$ and $\overline{DP}$ of the source set *S* and destination set *D*, respectively. In the following definition these two partitions are defined.

Definition 4.1

Let $\overline{SP} = \{S_k : S_k \subseteq S\}$ $\left(\overline{DP} = \{D_l : D_l \subseteq D\}\right)$ be a partition of the set of sources (destinations) satisfying

(i) $\bigcup_{S_k \in \overline{SP}} S_k = S$

(ii) $S_i \cap S_j = \varnothing \;\; \forall i \neq j;\; S_i, S_j \in \overline{SP}$

(the same must hold for $\overline{DP}$)

The following figure shows a partition of the set of sources *S* and the set of destination *D*. For the partition of the source set we have $\overline{SP} = \{S_1, S_2\}$ with $S_1 = \{1,2,3\}$ and $S_2 = \{4\}$. The partitions of the destination set is given by $\overline{DP} = \{D_1, D_2\}$ with $D_1 = \{1,2\}$ and $D_2 = \{3,4\}$

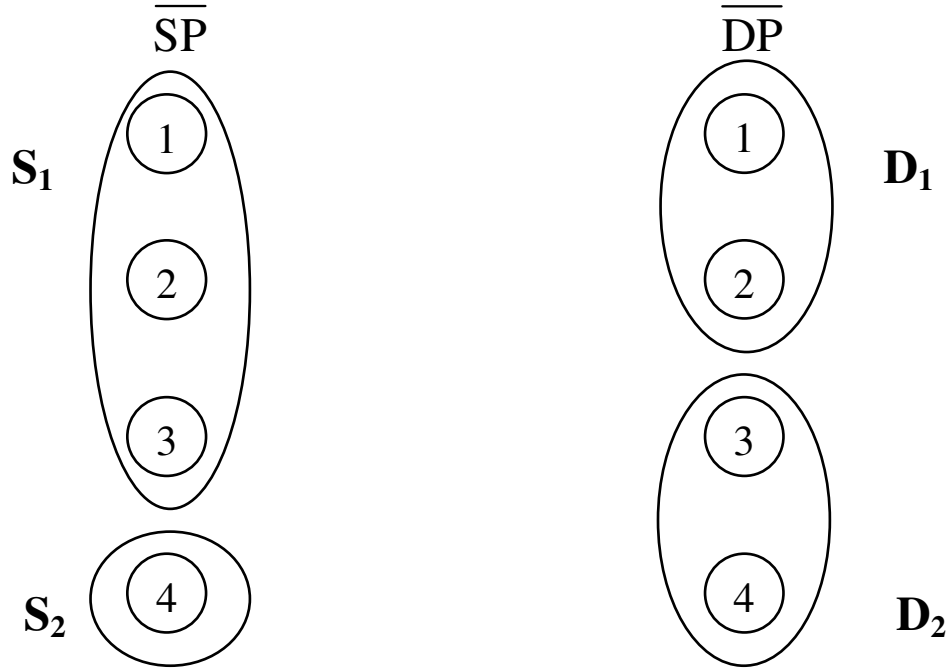


Figure 15: Partition of the set of sources and destination applied to the transportation problem of Example 4.1

Briefly spoken, after applying an aggregation, a source $k \in \overline{S}$ in the aggregated problem, replaces all nodes in the subset S_k. The arcs leaving the grouped source nodes of the original problem in direction to the same destination node are replaced through a single arc in the aggregated problem. The following figure shows the aggregated problem based on $\overline{SP}$ and $\overline{DP}$ defined above.

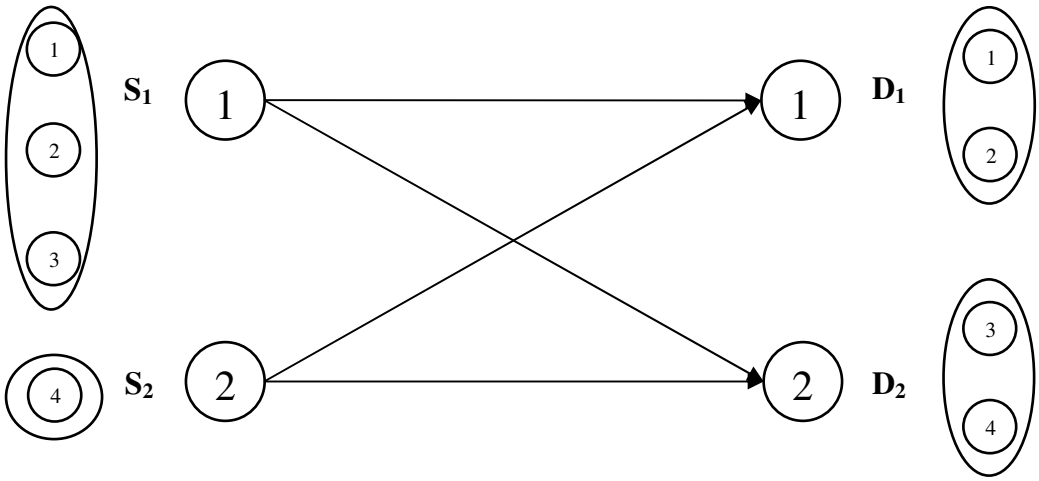


Figure 16: Aggregated transportation problem for the original problem of Example 4.1

Based on a partition of the source set *S* and the destination set *D*, we get the following formal description of the aggregated transportation problem.

The Aggregated Transportation Problem

(ATP)

$$\min \sum_{k\in\bar{S},l\in\bar{D}} \bar{y}_{kl}\bar{c}_{kl}$$

s.t.

$$\sum_{l\in\bar{D}} \bar{y}_{kl} = \bar{a}_k \qquad \forall k\in\bar{S}$$

$$\sum_{k\in\bar{S}} \bar{y}_{kl} = \bar{b}_l \qquad \forall l\in\bar{D}$$

$$\bar{y}_{kl} \geq 0 \qquad \forall k\in\bar{S}, l\in\bar{D}$$

Where:

$\bar{y}_{kl}$ = flow from source $k\in\bar{S}$ to destination $l\in\bar{D}$

$\bar{c}_{kl}$ = cost for shipping one unit of flow from source $k\in\bar{S}$ to destination $l\in\bar{D}$

$\bar{S} = \{k : S_k \in \overline{SP}\} = \{1,...,k,...,\bar{n}\}$ the set of sources

$\bar{D} = \{l : D_l \in \overline{DP}\} = \{1,...,l,...,\bar{m}\}$ the set of destinations

$\bar{a}_k = \sum_{s\in S_k} a_s$ the supply of source node $k\in\bar{S}$

$\bar{b}_l = \sum_{d\in D_l} b_d$ the demand of destination node $l\in\bar{D}$

Further let us denote with $\bar{y}^* = (\bar{y}^*_{11},...,\bar{y}^*_{kl},...,\bar{y}^*_{\bar{n}\bar{m}})$ any optimal solution for the ATP, with the corresponding objective value $\bar{z}^* = \sum_{k\in\bar{S},l\in\bar{D}} \bar{y}^*_{kl}\bar{c}_{kl}$.

For defining the costs $\bar{c}_{kl}$ of the aggregated problem, different kinds of *respecification maps* exist. Balas [Bal65] suggested a method for defining costs, which is oriented on the *aggregation by dominance* approach, a method for aggregating general linear programs. He defined the cost $\bar{c}_{kl}$ as:

$$\bar{c}_{kl} = \min_{\substack{s \in S_k \\ d \in D_l}} c_{sd} \qquad k \in \bar{S}, l \in \bar{D}$$

The so-called *weighted aggregation* approach used by Zipkin [Zip80] uses a form of convex combination to derive the costs of the aggregated problem.

$$\bar{c}_{kl} = \sum_{s \in S_k} \sum_{d \in D_l} g_{sd}^{kl} c_{sd} \qquad k \in \bar{S}, l \in \bar{D}$$

Where:

$$g_{sd}^{kl} = g_s^k g_d^l$$

$$g_s^k = \frac{a_s}{\bar{a}_k}$$

$$g_d^l = \frac{b_d}{\bar{b}_l}$$

Clearly, the aggregation by dominance requires less effort to set up the aggregate problem than using the concepts of Zipkin. However, the properties of Zipkin's more involved method permit the derivation of a very simple *disaggregation map*.

Before we come to these concepts, we finish the problem formulation with some concluding definitions:

$$k(s) = \text{index } k \in \bar{S} \text{ such that } s \in S_k \,,\, s \in S$$

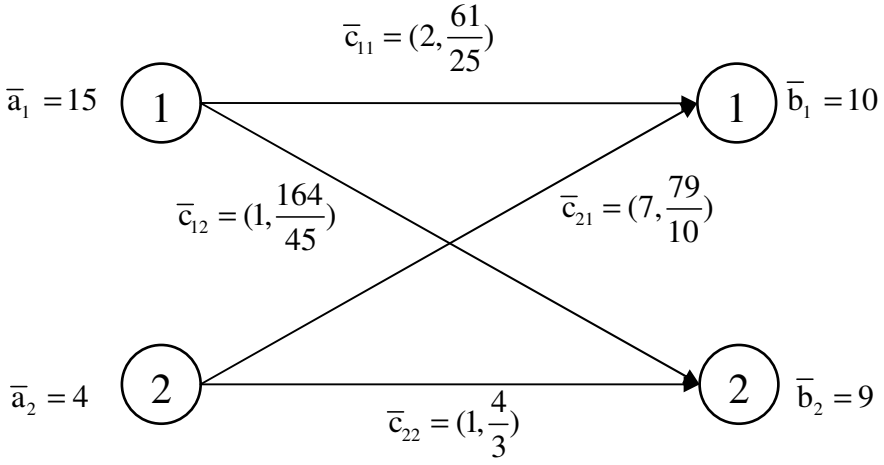
$$l(d) = \text{index } l \in \bar{D} \text{ such that } d \in D_l \,,\, d \in D$$

$$(\bar{u}, \bar{v}) = (\bar{u}_k, \bar{v}_l) = \text{an optimal solution of the dual of ATP}$$

In the following example we provide a complete example of an ATP, based on the UATP given in Example 4.1 and the partitions given in Figure 15, including the costs, supply and demand of the aggregated problem.

Example 4.2

$\overline{c}_{kl}$ = (aggregation by dominance, weighted aggregation)



4.2 The Algorithm of Balas

Balas' work on the aggregation of transportation problems is one of the earliest and most complete researches in this area. His algorithm is a good example that shows the possibility of how aggregation can be used for large scale transportation problems. The work of Balas can be seen as the basic concepts for the aggregation of network flow problems, which was used by many other authors as a starting point of their work. For example, Lee [Lee75] and Francis [Fra85] extended Balas' results, to general linear minimum cost network flow problems. Their concepts will be presented in the next chapter. Briefly spoken, Balas' algorithm, starts with the derivation of an aggregated problem based on the original problem, solves the APT to optimality and derives the so-called *partial problem*. The partial problem consists only of parts of the original problem, which are corresponding to flow variables greater than zero (basic variables) in the optimal solution of the ATP. The optimal solution to the partial problem serves as a fairly good initial solution for the original problem (UATP). Based on this initial solution, the regions of the original problem's cost matrix are detected, which worked upon in order to improve the given solution.
The efficiency of the presented algorithm increases with the size of the problem. However, it is not possible to give a general statement about the performance. Before presenting the algorithm, some definitions have to be given.

Note: The aggregated problems in this section are based on partitions $\overline{SP}$ and $\overline{DP}$. In order to derive the costs of the aggregated problem the approach of Balas is used (i.e. $\overline{c}_{kl} = \min_{\substack{s \in S_k \\ d \in D_l}} c_{sd} \qquad k \in \overline{S}, l \in \overline{D}$)

Definition 4.2

Given an UATP and a corresponding ATP based on a partition of *S* and *D*. Let $\bar{y} = (\bar{y}_{kl})$ be any feasible solution of the ATP and let $Q = \{(k,l) : k \in \bar{S}, l \in \bar{D}\}$ be the set of all arcs of the aggregated problem. Let us further define

$$\bar{R} = \{(k,l) \in Q : \bar{y}_{kl} > 0\}$$

$$\bar{P} = \{(s,d) : s \in S_k, d \in D_l; (k,l) \in \bar{R}\}$$

where $\bar{R}$ is the set of all arcs of the aggregated problem, which have flow non-zero and $\bar{P}$ is the set of all arcs of the original problem which are represented by arcs in $\bar{R}$.

Using these definitions we are now able to define the partial problem corresponding to a feasible solution $\bar{y}$ of the ATP:

Partial Transportation Problem

(PTP)

$$\min \sum_{(s,d)\in\bar{P}} \bar{x}_{sd} c_{sd}$$

s.t.

$$\sum_{d:(s,d)\in\bar{P}} \bar{x}_{sd} = a_s \qquad \forall s \in S$$

$$\sum_{s:(s,d)\in\bar{P}} \bar{x}_{sd} = b_d \qquad \forall d \in D$$

$$\bar{x}_{sd} \geq 0 \qquad \forall (s,d) \in \bar{P}$$

There should be raised the question if the partial problem has always a feasible solution ? This is a reasonable question because we do not have the same arcs as in the original problem. The following theorem gives an appropriate answer to this question. It is possible to show that a partial problem derived from a feasible solution $\bar{y}$ of the aggregated problem has always a feasible solution. In the proof a *disaggregation map* is defined (taken from Balas [Bal65]). It transforms the solution of the aggregated problem into a feasible solution of the partial problem. Obviously, this is also a feasible solution for the original problem.

Theorem 4.1 [Bal65]

Let $\bar{y}$ be a feasible solution for the ATP. Then there exists also a feasible solution $\bar{x} = (\bar{x}_{sd})$ for the corresponding partial problem PTP.

Proof:

Define $\bar{x}_{sd} = \left(\frac{a_s b_d}{\bar{a}_k \bar{b}_l}\right) \bar{y}_{kl} \qquad s \in S_k, d \in D_l; (k,l) \in \bar{R}$

To show: i) $\sum_{d:(s,d)\in\bar{P}} \bar{x}_{sd} = a_s \qquad \forall s \in S$

ii.) $\sum_{s:(s,d)\in\bar{P}} \bar{x}_{sd} = b_d \qquad \forall d \in D$

iii.) $\bar{x}_{sd} \geq 0 \qquad \forall (s,d) \in \bar{P}$

To i.)

$$\sum_{d:(s,d)\in\bar{P}} \bar{x}_{sd} = \sum_{d:(s,d)\in\bar{P}} \left(\frac{a_s b_d}{\bar{a}_{k(s)} \bar{b}_{l(d)}} \right) \bar{y}_{k(s)l(d)} = \sum_{l\in\bar{D}:(k(s),l)\in\bar{R}} \sum_{d\in D_l} \left(\frac{a_s b_d}{\bar{a}_{k(s)} \bar{b}_l} \right) \bar{y}_{k(s)l}$$

$$= \frac{a_s}{\bar{a}_{k(s)}} \sum_{l\in\bar{D}:(k(s),l)\in\bar{R}} \bar{y}_{k(s)l} \underbrace{\left(\frac{\sum_{d\in D_l} b_d}{\bar{b}_l} \right)}_{=1} = a_s \frac{\bar{a}_{k(s)}}{\bar{a}_{k(s)}} = a_s$$

To ii.)

Same proceeding as in i.)

To iii.)

$$\bar{x}_{sd} = \underbrace{\left(\frac{a_s b_d}{\bar{a}_{k(s)} \bar{b}_{l(d)} } \right)}_{>0} \overbrace{\bar{y}_{k(s)l(d)}}^{>0} > 0$$

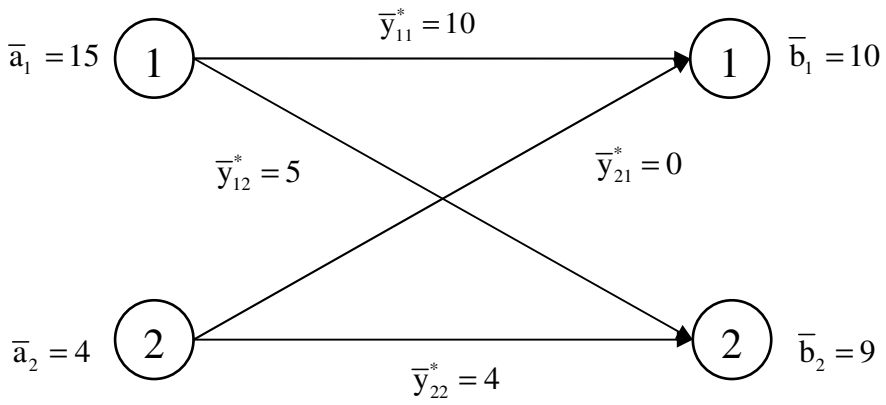
q.e.d.

Note: The solution derived through the disaggregation map will yield non-integer solutions for the original transportation problem in general.

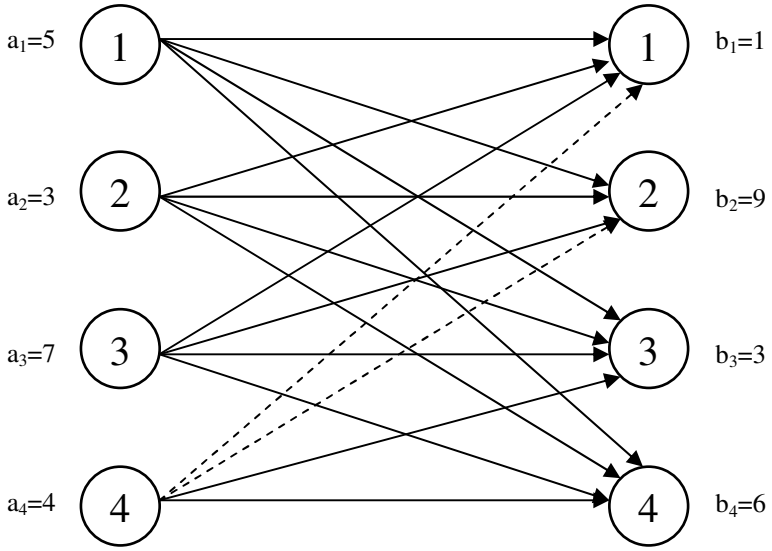
In the following example we see how the partial transportation problem (PTP) is derived from an (optimal) solution of the aggregated problem and we also see to what results the disaggregation map described in the proof above leads.

Example 4.3

Optimal solution to the aggregated problem of Example 4.2



The corresponding PTP (the dashed lines can be cancelled out)



Sour. / Dest.	1	2	3	4
1	1/3	3	5/9	10/9
2	1/5	9/5	1/3	2/3
3	7/15	21/5	7/9	14/9
4	0	0	4/3	8/3

Table IV: Feasible solution for the partial transportation problem of Example 4.3 applying the disaggregation map of Theorem 4.1

As mentioned above, only the flow between source 1 and destination 2 is integral.

The first steps of Balas' algorithm consist of the derivation of the ATP, solve the ATP to optimality and derive the corresponding PTP. The derived PTP is also solved to optimality. Based on the dual solution of the partial problem, the dual feasibility of the non-basic blocks of aggregated variables is evaluated. Therefore, the following definition involves a form of overestimating dual variables, which will be used in the algorithm.

Definition 4.3

Let $\bar{y}^*$ be an optimal solution to ATP and let $\bar{x}^*$ an optimal solution to the corresponding PTP. $(\tilde{u},\tilde{v})$ denotes the optimal dual pair, of the dual problem of the PTP. Then RC_{sd} is defined as

$$RC_{sd} = c_{sd} - \tilde{u}_s - \tilde{v}_d$$

Let us further define

$$\hat{u}_k = \max_{s \in S_k} \tilde{u}_s$$

$$\hat{v}_l = \max_{d \in D_l} \tilde{v}_d$$

$$\widehat{RC}_{kl} = \bar{c}_{kl} - \hat{u}_k - \hat{v}_l$$

$$V = \{(k,l) : (k,l) \in Q \setminus \bar{R}^*;\ \widehat{RC}_{kl} \geq 0\}$$

Note: $\bar{R}^*$ corresponds to the optimal solution $\bar{y}^*$ of the aggregated problem (see Definition 4.2)

By using the stated definitions, the following proposition is straightforward. The proposition will be used as the termination criterion in the algorithm.

Proposition 4.1 [Bal65]

An optimal solution $\bar{x}^*$ to the partial transportation problem is also optimal for the unaggregated transportation problem $\Leftrightarrow RC_{sd} \geq 0 \quad \forall s \in S_k, d \in D_l : (k,l) \in Q \setminus (\bar{R}^* \cup V)$

The necessity of the condition above is obvious. The sufficiency follows from two facts.

(I) Because $\bar{x}^*$ is an optimal solution for PTP, it holds that $RC_{sd} = 0 \; \forall \; \bar{x}^*_{sd} > 0$ and $RC_{sd} \geq 0 \; \forall \; s \in S_k, d \in D_l$ where $(k,l) \in \bar{R}^*$

(II) From the definition of V and $\widehat{RC}_{kl}$ we get that $RC_{sd} \geq 0$ for $s \in S_k, d \in D_l$ where $(k,l) \in V$

We are now able to present the algorithm of Balas.

Algorithm of Balas [Bal65] for solving large-scale transportation problems

INPUT: An unaggregated transportation problem (UATP)

OUTPUT: An optimal solution to UATP in a finite number of steps

STEP 1: Construct ATP based on a partition of the source set *S* and the destination set *D* of the original problem (UATP)

STEP 2: Solve the ATP to optimality and denote the optimal solution by $\bar{y}^*$

STEP 3: Construct the PTP based on $\bar{R}^*$ corresponding to $\bar{y}^*$

STEP 4: Solve PTP to optimality and denote the optimal solution by $\bar{x}^*$ and the optimal dual solution by $(\tilde{u}, \tilde{v})$

STEP 5: Compute $\widehat{RC}_{kl}$ for each $(k,l) \in Q \setminus \bar{R}^*$

If $\widehat{RC}_{kl} \geq 0 \; \forall (k,l) \in Q \setminus \bar{R}^*$ → **STOP** ($\bar{x}^*$ is also an optimal solution to UATP)

Else define $V' = Q \setminus \bar{R}^* \cup V$

STEP 6: Compute RC_{sd} for $s \in S_k, d \in D_l, (k,l) \in V'$
If $RC_{sd} \geq 0 \ \forall \ s \in S_k, d \in D_l; (k,l) \in V'$ → **STOP** ($\bar{x}^*$ is also an optimal solution to UATP)

STEP 7: Consider the new partial problem related to the set $\bar{R}^* \cup V'$

STEP 8: Solve the refined partial problem to optimality

STEP 9: Set $\bar{R}^* = \bar{R}^* \cup V'$ and GoTo Step 5,

Theorem 4.2 [Bal65]

The algorithm defined above leads to an optimal solution for the original problem (UATP) in a finite number of iterations.

We should mention that the algorithm given above may lead to the same complexity as solving the original problem directly.
We have not discussed how a partition of the source (destination) set should look like, so far. Of course, the selection of a partition is an important part concerning aggregation. On the other hand it is very difficult to give general advices. In most cases, the partitioning mainly depends on the given problem and its characteristics. In Chapter 5 we will see an algorithm for grouping nodes in a reasonable way which can be applied to the transportation problem, too. Even though Balas introduces some kind of measure/constraint for aggregating nodes

$$| \bar{c}_{kl} - c_{ds} | \leq \alpha \qquad s \in S_k, d \in D_l; k \in \bar{S}, l \in \bar{D}$$

he did not evaluate different values for α or gave further information on how to choose this value in an appropriate way. α is a problem dependent parameter and would represent an interesting value for an empirical sensitivity analysis. One general observation for the aggregation level of the original problem can be given so far: The more the size of the ATP is reduced (e.g. very high aggregation) the larger the related partial transportation problem will grow. This means that the effect of solving the ATP first, to get a fairly good initial solution for the original problems, will be smaller with growing scale of aggregation.
The main advantage of Balas' approach is that his algorithm finally leads to an optimal solution for the original problem. He only uses parts of the original data in the different steps of the algorithm. However, it would also be interesting to know how far we are at most from the optimal solution if we terminate the algorithm at an iteration *t*, for instance. Such information was not provided by Balas. In the next chapter, we will provide such a bound which can be used as an alternative termination criterion for the algorithm.

4.3 Zipkin's Weighted Aggregation Approach

As we have seen in the previous section, Balas aimed at an optimal solution for the original problem. Although his algorithm tries to use only parts of the complete data, it may result in an iteration in which we have to solve the original problem. Zipkin has a different point of view about aggregation. He derived bounds on the loss of accuracy, instead of solving the original transportation problem to optimality by using concepts of aggregation. The loss of accuracy is caused by solving the aggregated problem instead of the original one.
Zipkin's *weighted aggregation* and *fixed-weight disaggregation* permits the derivation of two a posterior and two a priori bounds. Zipkin mainly utilizes results from basic duality theory to derive these bounds. His approach can be summarized as follows: construct the aggregated transportation problem (ATP) based on a partition $\overline{SP}$ and $\overline{DP}$, solve the ATP to optimality and recover a feasible solution for the UATP by a very simple disaggregation map. Due to the particular respecification map used by Zipkin, the disaggregated solution for the original problem has the same objective value as the optimal solution of the ATP.
Therefore, it is possible to show that the optimal solution to the ATP is an upper bound for the original problem. It is clear that the loss of accuracy or the error caused by aggregation is the difference between the costs (the objective function value) of the approximate solution derived from the solution to the ATP and the costs of an optimal solution to the UATP. As mentioned above, Zipkin derived two kinds of bounds on this error. The a posterior bounds need the computation of an optimal solution for the ATP, whereas the a priori bounds can be calculated in advance, without solving a problem. Obviously, the a posterior bounds are tighter, because more information of the problem are included.
In the following, the concepts of Zipkin's weighted aggregation will be presented, which will finally result in the derivation of two a posterior and two a priori bounds for the loss of accuracy. In most cases the fixed-weight disaggregation results in a non integer solution for the original problem. Therefore, we will take up this problem by presenting an alternative aggregation method. The chapter will be concluded with a brief discussion about the differences between the concepts of Balas and Zipkin.

4.3.1 Weighted Aggregation

As we have seen in Section 4.1, the main differences between Balas and Zipkin concerning the aggregation lies in the definition of the respecification map for the costs. Zipkin used a kind of convex combination to derive the costs of the aggregated problem. In order to compute the costs of an aggregated arc he takes the sum over the weighted costs of the corresponding original arcs. Therefore, aggregation that uses such a respecification map can be termed as a *weighted aggregation*. So let us recall the definition of the respecification map for the costs:

$$\bar{c}_{kl} = \sum_{s \in S_k} \sum_{d \in D_l} g_{sd}^{kl} c_{sd} \quad k \in \bar{S}, l \in \bar{D}$$

Where:

$$g_{sd}^{kl} = g_s^k g_d^l$$

$$g_s^k = \frac{a_s}{\bar{a}_k}$$

$$g_d^l = \frac{b_d}{\bar{b}_l}$$

Note: All the aggregated problems derived in the following are based on a partition $\overline{SP}$ and $\overline{DP}$ as well as by using the respecification map of the weighted aggregation approach for deriving the costs.

4.3.2 Fixed-weight Disaggregation

Suppose the ATP has been solved and we are not only interested in an approximate objective value, but also do we need a feasible solution for the UATP. The so-called *disaggregation* recovers a feasible solution for the UATP, from the solution of the ATP. Depending on the definition of parameters (e.g. costs in the case of the transportation problem) different possibilities for disaggregation exist. One possible disaggregation method is the *fixed-weight disaggregation*; a method that allows a very quick and simple recovering of a feasible solution for the original problem. In the proof of Theorem 4.1 it is the first time the fixed-weight disaggregation map is mentioned in the literature, although Balas' algorithm does not specifically use this or any other disaggregation map. We will see in Paragraph 4.3.4 another more complex disaggregation method, which will lead to better solutions for the UATP. But before we come to this method we will have a closer look on the fixed-weight disaggregation in the following.

Definition 4.4

A solution $\overline{x}$ to the UATP is called a *fixed-weight solution*, if it is derived from a solution $\overline{y}$ of a corresponding ATP in the following way

$$\overline{x}_{sd} = g_{sd}^{kl}\, \overline{y}_{kl}\,; \quad s \in S_k, d \in D_l$$

Remark: As already mentioned before the fixed-weight solution was also used in the proof of Theorem 4.1, in which we already observed that the derived solution for the original problem is not integer in general (e.g. see Example 4.3).

In the following proposition we will see that the fixed-weight solution is indeed a feasible solution for the original problem. We will also show that the objective value corresponding to the fixed-weight solution is equal to the optimal objective value of the ATP.

Proposition 4.2 [Zip80]

(a) Let $\overline{y}$ be feasible for ATP and $\overline{x}$ the corresponding fixed-weight solution.

$$\Rightarrow \overline{x} \text{ is a feasible solution for the UATP}$$

(b) Let $\overline{y}^*$ be optimal for ATP and $\overline{x}^*$ the corresponding fixed-weight solution.

$$\Rightarrow \sum_{s \in S, d \in D} c_{sd} \overline{x}_{sd}^* = \overline{z}^*$$

The proof of part a.) stated below uses the same concepts already seen in the proof of Theorem 4.1, but adjusted to the setting of the weighted aggregation.

Proof:

a.) To show: i.) $\sum_{d\in D} \bar{x}_{sd} = a_s \qquad \forall s \in S$

ii.) $\sum_{s\in S} \bar{x}_{sd} = b_d \qquad \forall d \in D$

iii.) $\bar{x}_{sd} \geq 0 \qquad \forall s \in S, \forall d \in D$

To i.)

$$\sum_{d\in D} \bar{x}_{sd} = \sum_{d\in D} g_{sd}^{k(s)l(d)}\, \bar{y}_{k(s)l(d)} = \sum_{d\in D} g_s^{k(s)}\, g_d^{l(d)}\, \bar{y}_{k(s)l(d)} = g_s^{k(s)} \sum_{d\in D} g_d^{l(d)}\, \bar{y}_{k(s)l(d)}$$

$$= g_s^{k(s)} \sum_{l\in \bar{D}} \bar{y}_{k(s)l} \sum_{d\in D_l} g_d^l = g_s^{k(s)} \sum_{l\in \bar{D}} \bar{y}_{k(s)l} \sum_{d\in D_l} \frac{b_d}{\bar{b}_l} = g_s^{k(s)} \sum_{l\in \bar{D}} \bar{y}_{k(s)l} \frac{\sum_{d\in D_l} b_d}{\bar{b}_l}$$

$$= g_s^{k(s)} \sum_{l\in \bar{D}} \bar{y}_{k(s)l} = g_s^{k(s)}\, \bar{a}_{k(s)} = \frac{a_s}{\bar{a}_{k(s)}}\, \bar{a}_{k(s)} = a_s$$

To ii.) analogous to i.)

To iii.) $$\bar{x}_{sd} = g_{sd}^{kl}\, \bar{y}_{kl} = \underbrace{\left(\frac{a_s}{\bar{a}_k}\right)}_{>0} \overbrace{\left(\frac{b_d}{\bar{b}_l}\right)}^{>0} \underbrace{\bar{y}_{kl}}_{\geq 0} \geq 0$$

b.)

$$\sum_{s\in S, d\in D} c_{sd} \bar{x}_{sd}^* = \sum_{s\in S, d\in D} c_{sd} g_{sd}^{k(s)l(d)}\, \bar{y}_{k(s)l(d)}^*$$

$$= \sum_{k\in \bar{S}} \sum_{l\in \bar{D}} \bar{y}_{kl}^* \underbrace{\sum_{s\in S_k} \sum_{d\in D_l} c_{sd} g_{sd}^{kl}}_{\bar{c}_{kl}}$$

$$= \sum_{k\in \bar{S}} \sum_{l\in \bar{D}} \bar{y}_{kl}^* \bar{c}_{kl} = \bar{z}^*$$

q.e.d.

Remark: Proposition 4.2 b.) does not hold in general.

The following result is straightforward because the fixed-weight solution is a feasible solution for the original problem, with objective value equal $\bar{z}^*$.

Corollary 4.1

$$\bar{z}^* \geq z^*$$

4.3.3 Bounds for the Loss of Accuracy

Let us assume we solved the aggregated problem to optimality and derived the corresponding fixed-weight solution $\bar{x}^*$. The objective value of the fixed-weight solution is equal to the optimal objective value of the aggregated problem $\bar{z}^*$. So the loss of accuracy induced by solving the aggregated problem instead of the original problem is given by $\bar{z}^* - z^*$. It would be useful to have some bounds on this latter quantity.
Therefore, in the following two a posterior and two a priori bounds are presented, which were derived by Zipkin. The latter ones have some interesting interpretation regarding the choice of partitions. But first we have to give a general bound, in order to derive the a posteriori and a priori bounds.

Proposition 4.3 [Zip80]

Let $\bar{y}^*$ be an optimal solution for ATP and $(\bar{u},\bar{v})$ the corresponding optimal dual solution. Then the following inequality holds:

$$\bar{z}^* - z^* \le -\min_x \sum_{s\in S, d\in D} (c_{sd} - \bar{u}_{k(s)} - \bar{v}_{l(d)}) x_{sd} \qquad (4.1)$$

s.t.

$$\sum_{d\in D} x_{sd} = a_s \qquad \forall s \in S$$
$$\sum_{s\in S} x_{sd} = b_d \qquad \forall d \in D$$
$$x_{sd} \ge 0 \qquad \forall s \in S, \forall d \in D$$

First of all we want to give two observations concerning the derived error bound, before we start with the proof.

1.) The derived error bound has a major drawback. Calculating the bound means that we have to solve

$$-\min_x \sum_{s\in S, d\in D} \underbrace{(c_{sd} - \bar{u}_{k(s)} - \bar{v}_{l(d)})}_{e_{sd}} x_{sd} = -\min_x \sum_{s\in S, d\in D} e_{sd}\, x_{sd}$$

s.t.

$$\sum_{d\in D} x_{sd} = a_s \qquad \forall s \in S$$
$$\sum_{s\in S} x_{sd} = b_d \qquad \forall d \in D$$
$$x_{sd} \ge 0 \qquad \forall s \in S, \forall d \in D$$

which has the same complexity and problem size as solving the UATP.

2.) You can think of this expression as a kind of weak duality result for a form of generalized dual of the UATP. Indeed the dual variables of UATP are presented in a generalized form.

The following proof is based on the concepts of Zipkin [Zip80]

Proof of Proposition:

$$z^* = \sum_{s\in S,d\in D} c_{sd}x^*_{sd} = \sum_{s\in S,d\in D} c_{sd}x^*_{sd} + \sum_{s\in S}\bar{u}_{k(s)}\overbrace{\left(a_s - \sum_{d\in D} x^*_{sd}\right)}^{=0} + \sum_{d\in D}\bar{v}_{l(d)}\overbrace{\left(b_d - \sum_{s\in S} x^*_{sd}\right)}^{=0}$$

$$= \sum_{s\in S,d\in D} c_{sd}x^*_{sd} + \sum_{s\in S}\bar{u}_{k(s)}a_s - \sum_{s\in S}\bar{u}_{k(s)}\sum_{d\in D} x^*_{sd} + \sum_{d\in D}\bar{v}_{l(d)}b_d - \sum_{d\in D}\bar{v}_{l(d)}\sum_{s\in S} x^*_{sd}$$

$$= \sum_{s\in S,d\in D} c_{sd}x^*_{sd} + \sum_{k\in\bar{S}}\bar{u}_k \underbrace{\sum_{s\in S_k} a_s}_{\bar{a}_k} - \sum_{s\in S}\sum_{d\in D}\bar{u}_{k(s)}x^*_{sd} + \sum_{l\in\bar{D}}\bar{v}_l \underbrace{\sum_{d\in D_l} b_d}_{\bar{b}_d} - \sum_{d\in D}\sum_{s\in S} x^*_{sd}\bar{v}_{l(d)}$$

$$= \sum_{s\in S,d\in D} c_{sd}x^*_{sd} + \sum_{k\in\bar{S}}\bar{u}_k\bar{a}_k + \sum_{l\in\bar{D}}\bar{v}_l\bar{b}_l - \sum_{s\in S}\sum_{d\in D}\bar{u}_{k(s)}x^*_{sd} - \sum_{d\in D}\sum_{s\in S} x^*_{sd}\bar{v}_{l(d)}$$

$$= \underbrace{\sum_{k\in\bar{S}}\bar{u}_k\bar{a}_k + \sum_{l\in\bar{D}}\bar{v}_l\bar{b}_d}_{\bar{z}^*} + \sum_{s\in S,d\in D}(c_{sd} - \bar{u}_{k(s)} - \bar{v}_{l(d)})x^*_{sd}$$

$$= \bar{z}^* + \sum_{s\in S,d\in D}(c_{sd} - \bar{u}_{k(s)} - \bar{v}_{l(d)})x^*_{sd}$$

Therefore we get that:

$$z^* = \bar{z}^* + \sum_{s\in S,d\in D}(c_{sd} - \bar{u}_{k(s)} - \bar{v}_{l(d)})x^*_{sd}$$

So we finally get that:

$$\bar{z}^* - z^* \le -\min_x \sum_{s\in S,d\in D}(c_{sd} - \bar{u}_{k(s)} - \bar{v}_{l(d)})x_{sd}$$

s.t. *x* satisfies the constraints of the original transportation problem.

q.e.d.

Although it would make no sense to use the derived bound directly, it can be considered as a basis for further bounds. Since any relaxation of the problem in (4.1), however, also yields a valid bound. So let us have a look at such a relaxation. Dropping the supply respective the demand constraints will lead us to the following bounds.

A posterior bounds for the loss of accuracy [Zip80]

$$\overline{z}^* - z^* \le \sum_{d \in D} b_d \max_{s \in S} (\overline{u}_{k(s)} + \overline{v}_{l(d)} - c_{sd}) \qquad (4.2)$$

$$\overline{z}^* - z^* \le \sum_{s \in S} a_s \max_{d \in D} (\overline{u}_{k(s)} + \overline{v}_{l(d)} - c_{sd}) \qquad (4.3)$$

Bound (4.2) is derived by dropping the supply and bound (4.3) is derived by dropping the demand constraints. The computation of these bounds requires only a solution for the ATP or it's dual. Therefore, the complexity in computing this bound is considerably less than the work for bound (4.1). As Zipkin, we did not succeed in evaluating which bound is tighter. However, limited numerical experience we made suggests that the bound in (4.2) is tighter when destinations are higher aggregated then sources, whereas for the bounds in (4.3) the opposite holds.

Based on this two a posterior bounds it is possible to derive two corresponding a priori ones.

Using the fact that

$$\overline{u}_k + \overline{v}_l \le \overline{c}_{kl} \qquad \forall k \in \overline{S}, l \in \overline{D}$$

holds, which is true, because $(\overline{u}, \overline{v})$ is a feasible (optimal) solution to the dual of the ATP, we get the following a priori bounds.

A priori bounds for the loss of accuracy [Zip80]

$$\overline{z} - z^* \le \sum_{d \in D} b_d \max_{s \in S} (\overline{c}_{k(s)l(d)} - c_{sd}) \qquad (4.4)$$

$$\overline{z} - z^* \le \sum_{s \in S} a_s \max_{d \in D} (\overline{c}_{k(s)l(d)} - c_{sd}) \qquad (4.5)$$

We can observe that, the more similar we choose the entities aggregated together, the lower the a priori bounds will be. Hence the bounds strongly depend on the partition of the set of sources and the set of destinations, respectively. Therefore, the bounds can be interpreted as a kind of measurement for the dissimilarity within the groups of sources and destinations aggregated together. Suppose for a moment that for each D_l the costs on corresponding arcs from each source are almost identical and let us further suppose that for each S_k the costs on corresponding arcs to each destination are also almost identical. We would see that each of the maximands $(\overline{c}_{k(s)l(d)} - c_{sd})$ will be close to zero. Each maximum will be close to zero, too and hence the bounds themselves will be very small. This interpretation conforms to the intuitive idea that aggregation of very similar nodes should result in less error.

Because of the fact that the a priori bounds are derived through an estimate of the a posterior bounds, it is obvious that the a posterior bounds are at least as good as the a priori bounds.

Let us calculate the different bounds for our example of the beginning of this chapter, before having a closer look on how it is possible to derive integer results from the disaggregation.

Example 4.4

Optimal flow for the original problem of Example 4.1:

Sour./Dest.	1	2	3	4
1	1	1	3	0
2	0	3	0	0
3	0	5	0	2
4	0	0	0	4

Table V: Flow of an optimal solution for the UATP defined in Example 4.1

With corresponding objective value $z^* = 34$

Optimal flow for the aggregated problem, with the corresponding optimal dual pair:

Sour./Dest.	1	2	$\overline{u}_k$
1	10	5	3,64
2	0	4	1,33
$\overline{v}_l$	-1,2	0	

Table VI: Flow of an optimal solution for the corresponding ATP

With corresponding objective value $\overline{z}^* = 47,96$

Finally we get the following results for the derived bounds:

	Maximands of the bounds $(\overline{u}_{k(s)} + \overline{v}_{l(d)} - c_{sd})$				Result
Bound (4.2)	0,4	0,4	2,6	0,6	*15,4*
Bound (4.3)	2,6	0,4	2,6	0,3	*33,6*
Bound (4.4)	0,9	0,44	2,64	0,64	*16,62*
Bound (4.5)	2,64	0,44	2,64	0,9	*36,6*

Table VII: The maximands required for the bounds 4.2-4.5. In the last column the particular value for the bounds can be found.

Taking the results derived above the following inequality holds:

$$\overline{z}^* - z^* = 47,96 - 34 = 13,96 \le \sum_{d \in D} b_d \max_{s \in S} (\overline{u}_{k(s)} + \overline{v}_{l(d)} - c_{sd}) = 15,4$$

4.3.4 The all-integer Disaggregation Method

The disaggregation method that has been used so far has a major drawback; the derived solution for the original problem is not integer in general. Zipkin [Zip77] presented another disaggregation method, specified by *optimal disaggregation*, which has an appealing property. If only destinations are aggregated and if the original data is integer, the feasible solution produced by disaggregating the ATP's solution itself is integer. We will not discuss this idea in our work, because an extension to this method was presented in a short communication of Raimer and Zipkin [RZ83]. It recovers an all-integer solution when both sources and destinations are aggregated. We will refer to this method as the *all-integer disaggregation* method. The bounds derived in the previous paragraph can also be applied for this kind of disaggregation. Before we show this, let us start with an algorithm for the all-integer disaggregation based on the ideas of Raimer and Zipkin.

All-integer Disaggregation Algorithm

INPUT: UATP with integral supply and demand; an optimal solution $\bar{y}^*$ for the ATP

OUTPUT: $\bar{x}'$ an integral solution to UATP derived from $\bar{y}^*$

STEP 1: For each source $k \in \bar{S}$
solve a transportation problem with the following settings:

Sources	→	$s \in S_k$
Destinations	→	$\bar{D}'_k = \{ l \in \bar{D} : \bar{y}^*_{kl} > 0 \}$
Supply	→	$a_s, s \in S_k$
Demand	→	$\bar{y}^*_{kl}, l \in \bar{D}'_k$
Costs	→	$\bar{c}_{sl} = \sum_{d \in D_l} g^l_d c_{sd},\ s \in S_k, l \in \bar{D}'_k$

Let us denote the corresponding optimal solution by $e' = (e'_{sl})$ where $e'_{sl} = 0$, if $s \in S_k$ and $l \notin \bar{D}'_k$ (*i.e.* $\bar{y}^*_{kl} = 0$)

STEP 2: For each destination $l \in \bar{D}$
solve a transportation problem with the following settings:

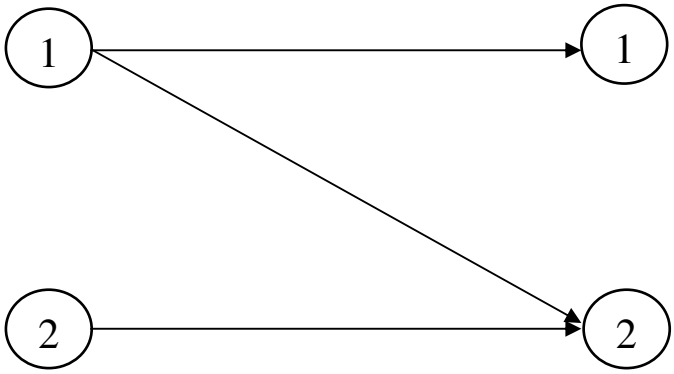
Sources	→	$S'_l = \{ s \in S : e'_{sl} > 0 \}$
Destinations	→	$d \in D_l$
Supply	→	$e'_{sl}, s \in S'_l$
Demand	→	$b_d, d \in D_l$
Costs	→	$c_{sd}, s \in S'_l, d \in D_l$

Let us denote the corresponding optimal solution by $\bar{x}' = (\bar{x}'_{sd})$ where $\bar{x}'_{sd} = 0$, if $d \in D_l$ and $s \notin S'_l$.

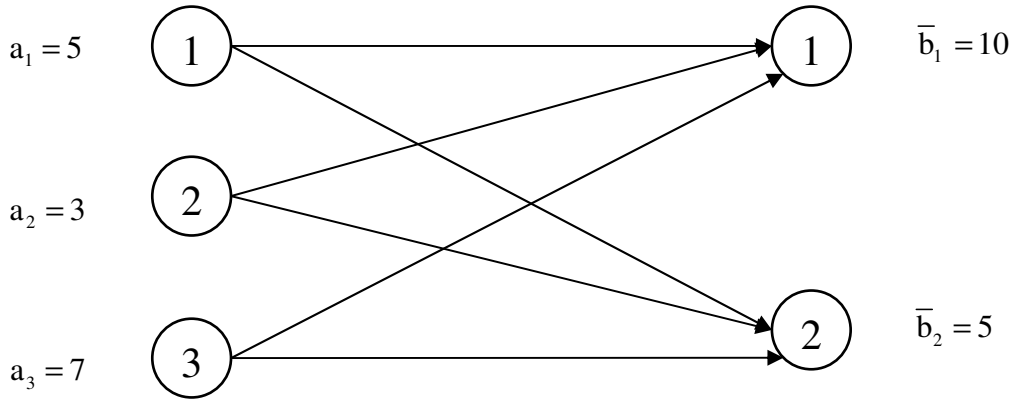
Before we show that $\bar{x}'$ is a feasible solution to UATP we want you to have a look at another example.

Example 4.5

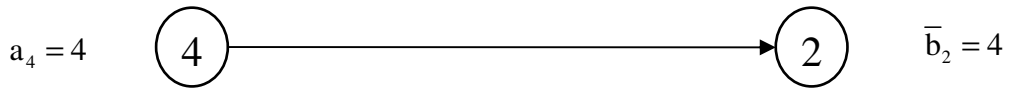
As we have seen in Example 4.3 the optimal flow of the ATP uses the following arcs.



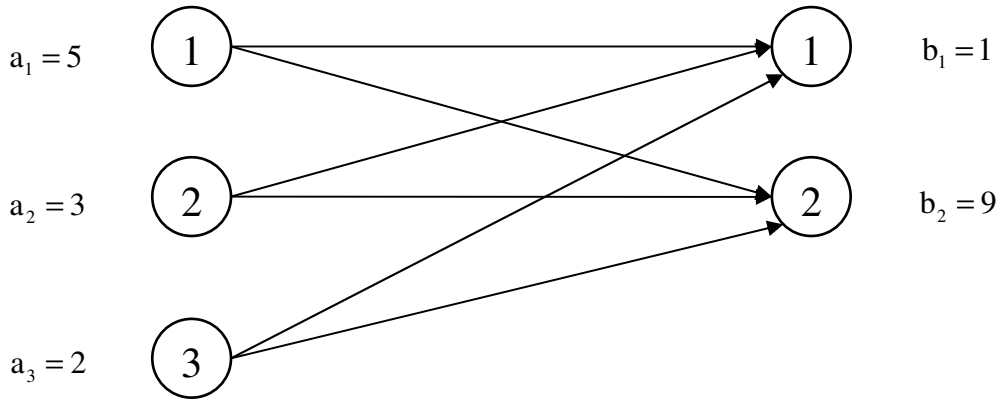
Hence the first transportation problem of Step 1 can be stated as follows:



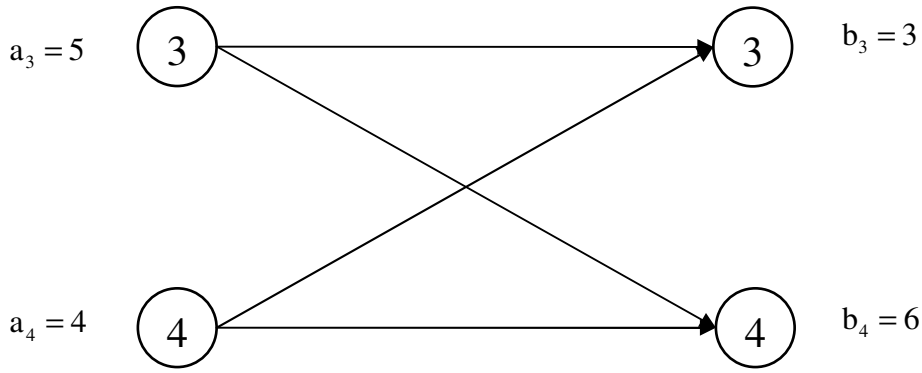
and the second transportation problem of Step 1 as follows:



Corresponding to the solutions of Step 1 we get the following first transportation problem of Step 2:



and the second problem of Step 2:



At the end we get a feasible, integral solution for the UATP with an objective value of **43**, which is less than the objective value of an optimal solution to ATP (i.e. $\overline{z}^* = 47,96$)

Proposition 4.4 [RZ83]

$\overline{x}'$ generated by the all-integer disaggregation method leads to a feasible, integral solution for the UATP.

Proof:

To show:

i.) $\sum_{d \in D} \overline{x}'_{sd} = a_s \qquad \forall s \in S$

ii.) $\sum_{s \in S} \overline{x}'_{sd} = b_d \qquad \forall d \in D$

iii.) $\overline{x}'_{sd} \geq 0 \qquad \forall s \in S, \forall d \in D$

vi.) x' is integral

To i.)

$$\sum_{d\in D} \overline{x}'_{sd} = \sum_{l\in \overline{D}} \sum_{d\in D_l} \overline{x}'_{sd} \overset{*}{=} \sum_{l\in \overline{D}} e'_{sl} \overset{**}{=} a_s$$

* since $\overline{x}'_{sd}$ is feasible for the l'th transportation problem in STEP 2
** since e'_{sd} is feasible for the k(s)'th transportation problem in STEP 1

To ii.) Clear (see the constraints of the transportation problems in STEP 2)

To iii.) Obvious, since the constraint is part of all sub problems

To iv.) If the original problem has integral supply and demand data, each of the sub problems has integral input data, too. Hence every single solution for the different problems is integer. So at the end the output of the algorithm is also integer.

q.e.d.

The result of the all-integer disaggregation method leads to an integral solution, which is needed by most real world applications. But what is about the bounds derived in the previous paragraph ? Are they still valid for this type of disaggregation ? Do we have to derive new bounds ? Do not hesitate to use the results derived so far, for the all-integer integration, too. The observation of Example 4.5, that the objective value corresponding to the all-integer solution is less than the optimal objective value of a solution to ATP, holds in general.

Proposition 4.5[RZ83]

Let $\overline{y}^*$ be an optimal solution to the ATP, $\overline{x}^*$ the corresponding fixed-weight solution and $\overline{x}'$ the corresponding disaggregated solution for the UATP generated by the all-integer method. Then the following inequality holds

$$\sum_{s\in S, d\in D} c_{sd} \overline{x}'_{sd} \leq \sum_{s\in S, d\in D} c_{sd} \overline{x}^*_{sd}$$

Before we proof the proposition, let us state the following corollary, which is a direct consequence of Proposition 4.5. It shows us that the results we got so far are also valid for the all-integer disaggregation.

Corollary 4.2:

The error bounds derived in the previous section, namely (4.1),(4.2), (4.3), (4.4) and (4.5) are valid for the solution generated by the all-integer disaggregation, too.

Proof of Proposition based on the basic ideas of [RZ83]:

Define $\overline{e}_{sl} = \sum_{d\in D_l} \overline{x}^*_{sd} = \sum_{d\in D_l} g_s^{k(s)} g_d^l \overline{y}^*_{k(s)l} = g_s^{k(s)} \overline{y}^*_{k(s)l} \sum_{d\in D_l} g_d^l = g_s^{k(s)} \overline{y}^*_{k(s)l};\ s\in S, l\in \overline{D}$

Claim 1: $(\overline{e}_{sl})_{s\in S_k, l\in \overline{D}'_k}$ is feasible for the k – th transportation problem of STEP 1

Proof of Claim 1:

To show: i) $\sum_{l\in\bar{D}_k'} \bar{e}_{sl} = a_s \qquad \forall s \in S_k$

ii.) $\sum_{s\in S_k} \bar{e}_{sl} = \bar{y}^*_{kl} \qquad \forall l \in \bar{D}_k'$

To i.) $$\sum_{l\in\bar{D}_k'} \bar{e}_{sl} = \sum_{l\in\bar{D}_k'} g_s^k \bar{y}^*_{kl} = g_s^k \sum_{l\in\bar{D}_k'} \bar{y}^*_{kl} = g_s^k \bar{a}_k = \frac{a_s}{\bar{a}_k}\bar{a}_k = a_s$$

To ii.) $$\sum_{s\in S_k} \bar{e}_{sl} = \sum_{s\in S_k} g_s^k \bar{y}^*_{kl} = \bar{y}^*_{kl} \sum_{s\in S_k} g_s^k = \bar{y}^*_{kl} \sum_{s\in S_k} \frac{a_s}{\bar{a}_k} = \bar{y}^*_{kl} \frac{\sum_{s\in S_k} a_s}{\sum_{s\in S_k} a_s} = \bar{y}^*_{kl}$$

q.e.d. (Claim 1)

Now let us define $x''_{sd} = g_d^{l(d)} e'_{sl(d)} \qquad s \in S, d \in D$

Claim 2: $(x''_{sd})_{s\in S_l', d\in D_l}$ is feasible for the l'th transportation problem of STEP 2

Proof of Claim 2:

To show: i.) $\sum_{d\in D_l} x''_{sd} = e'_{sl} \qquad \forall s \in S_l'$

ii.) $\sum_{s\in S_l'} x''_{sd} = b_d \qquad \forall d \in D_l$

To i.) $$\sum_{d\in D_l} x''_{sd} = \sum_{d\in D_l} g_d^l e'_{sl} = e'_{sl} \sum_{d\in D_l} \frac{b_d}{\bar{b}_l} = e'_{sl} \frac{\sum_{d\in D_l} b_d}{\sum_{d\in D_l} b_d} = e'_{sl}$$

To ii.) $$\sum_{s\in S_l'} x''_{sd} = \sum_{s\in S_l'} g_d^l e'_{sl} = g_d^l \sum_{s\in S_l'} e'_{sl} = g_d^l \sum_{k\in\bar{S}} \underbrace{\sum_{s\in S_k} e'_{sl}}_{=\bar{y}^*_{kl}} = \frac{b_d}{\bar{b}_l}\bar{b}_l = b_d$$

q.e.d. (Claim 1)

So coming to the proof of the inequality.

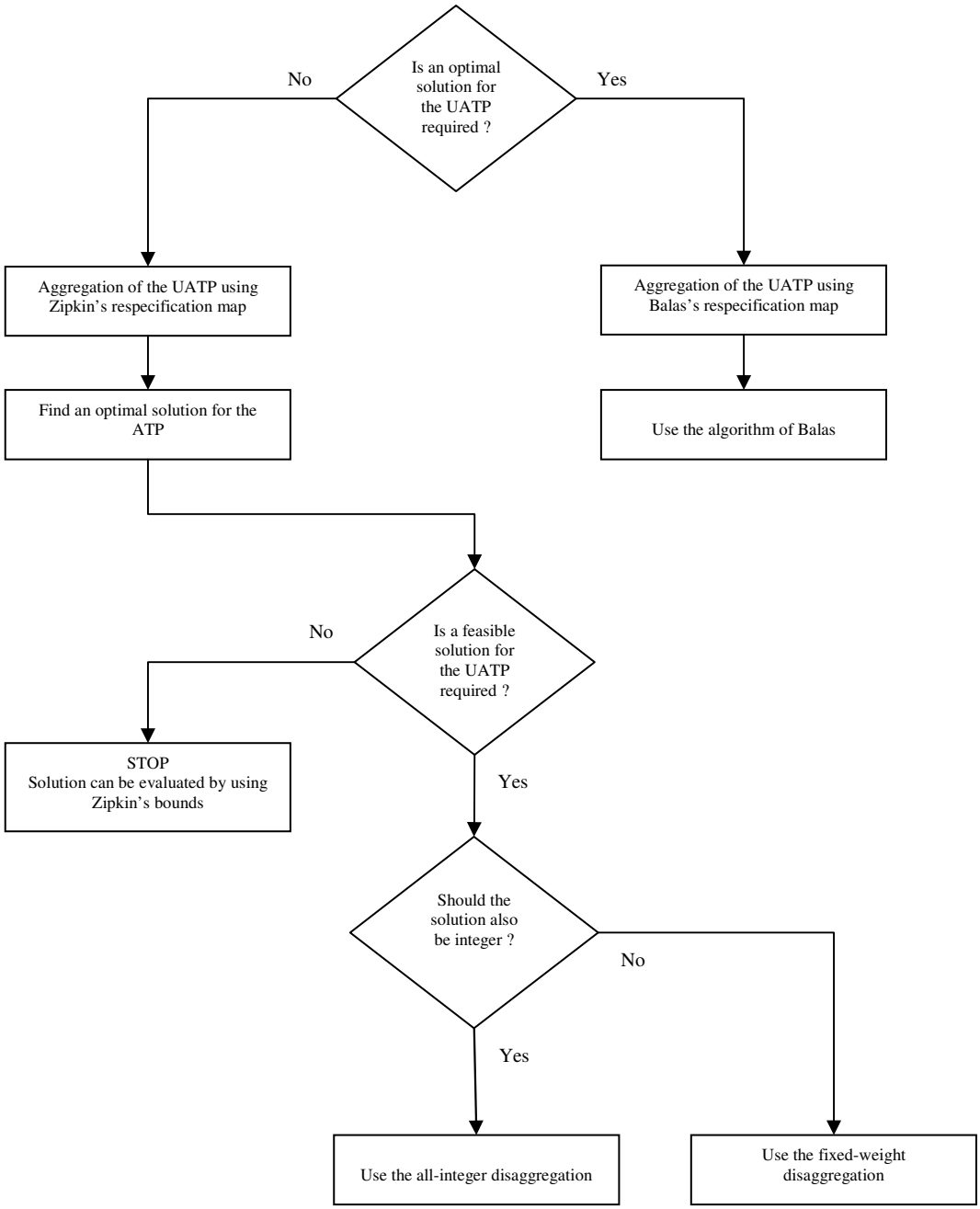
$$\sum_{s\in S, d\in D} c_{sd}\bar{x}'_{sd} \overset{\text{Claim 2}}{\leq} \sum_{s\in S, d\in D} c_{sd} x''_{sd} = \sum_{s\in S, d\in D} c_{sd} g_d^{l(d)} e'_{sl(d)} = \sum_{s\in S}\sum_{l\in\bar{D}} e'_{sl} \sum_{d\in D_l} c_{sd} g_d^l = \sum_{s\in S}\sum_{l\in\bar{D}} \bar{c}_{sl} e'_{sl}$$

$$\overset{\text{Claim 1}}{\leq} \sum_{s\in S}\sum_{l\in\bar{D}} \bar{c}_{sl}\bar{e}_{sl} = \sum_{s\in S}\sum_{l\in\bar{D}}\sum_{d\in D_l} c_{sd} \underbrace{g_s^{k(s)} g_d^l \bar{y}^*_{k(s)l}}_{\bar{x}^*_{sd}} = \sum_{s\in S, d\in D} c_{sd}\bar{x}^*_{sd}$$

q.e.d

4.4 Conclusion

As we have seen in the previous section, the all-integer disaggregation leads to a better solution for the original problem than the fixed-weight disaggregation. However, in order to derive the all-integer solution, we have to invest more effort. We have to solve several "sub" transportation problems, whereas the derivation of the fixed-weight disaggregation can be done very quickly. Which one of these two methods should be taken mainly depends on the underlying application. Because of the fact that this is also valid for the choice between Balas' and Zipkin's approach, we derived a flow chart, which provides an overview about the different situations which can occur, concerning the solution of a large-scale transportation problem and how we can react to them.



Let us close this section with some concluding statements. We still have to answer the question how “good” partitions of the source and destination set, respectively, can be generated. Therefore, we suggest reading the next chapter about the aggregation of minimum cost network flow problems. In it we will discuss, among other things, how the grouping of nodes can be arranged. Most of the ideas presented there are extensions of methods for the grouping of nodes of transportation problems. Therefore, it will be straightforward to specialize these ideas to transportation problems. However, we can already say that, even though it will be possible to derive heuristics for the grouping of nodes, the grouping will still depend on the particular application to a very high degree.

Chapter 5

Aggregation of the Minimum Cost Network Flow Problem

In Chapter 4, we discussed the basic approaches of aggregation and disaggregation applied to the transportation problem. There we presented the algorithm of Balas and the concepts of Zipkin. In the following chapter we want to extend these concepts to large-scale minimum cost network flow problems (MCNFP). They are more general then the transportation problems. Besides supply and demand nodes, already known from the transportation problem, we also have transshipment nodes. They can be interpreted as a kind of transfer station which has neither a supply nor a demand. Further we have lower- (which can be zero) and upper (which can be infinity) bounds on the flow on the arcs, as well as in the transportation problem costs for transportation. The objective is to satisfy the demand at a minimal cost without violating the bound and flow constraints, respectively. The classical transportation problem of linear programming, defined in the last chapter, is a minimum cost network flow problem without any transshipment nodes and no upper or lower bounds on arc capacities.
Minimum cost network flow problems can be found in almost all industries; applications rise from medical diagnosis (e.g. X-Ray Projection), to transportation planning or human resource management (e.g. scheduling).
Because of the fact that minimum cost network flow problems are linear problems, it is not surprising to discover that we can also use linear programming methodologies to solve them. Indeed we will see in the current chapter how basic duality theory can be used to derive bounds on the error caused by aggregation. The primal-dual or the out-of-kilter are very efficient, pseudo polynomial, algorithms for solving the MCNFP. Scaling algorithms also provide an optimal solution, but with a polynomial bound on the complexity. For a detailed overview about algorithms and theory concerning network flow problems including the MCNFP we refer to [AMO93]

As we begin to study aggregation for a more general problem, we should raise some questions:

1. Are there differences between the aggregation of a minimum cost network flow problem and the aggregation of a transportation problem ?

2. Can the theory stated so far be enhanced for the more general case of minimum cost network flow problems ?

3. How reasonable is an aggregation, for solving large-scale minimum cost network flow problems ?

In the following chapter, we will address these questions. Therefore, the work of three authors, namely Francis [Fra85], Lee [Lee75] and Zipkin [Zip77],[Zip80] is presented and extended. The work of Francis and Lee is based on the ideas of Balas, resulting also in an algorithm for large-scale minimum cost network flow problems, whereas the work of Zipkin

acts on the concepts he already introduced for the transportation problem. It is also possible to distinguish between the *aggregation by dominance* and the *weighted aggregation* as we have already seen for the transportation problem.
In the following section we will start with a problem description and definitions concerning aggregation which are valid for the approach of Lee/Francis as well as for the one of Zipkin. In Section 5.2 the algorithm of Lee will be discussed which was extended by Francis. Their ideas are based on the aggregation by dominance, resulting in an aggregated problem which is a relaxation of the original one. In Section 5.3 we discuss the weighted aggregation used by Zipkin, which finally results in a feasible solution for the original problem. As for the transportation problem bounds are derived to verify the goodness of this solution. The chapter will be closed with a section about measurements for the scale of aggregation and a discussion how nodes can be aggregated in a reasonable way.

5.1 Problem Description

Because of the fact that all three authors use different formulations for the MCNFP as well as different notations for aggregation, it seems reasonable to establish one common formulation and notation which will be used for the remainder of this work.

Unaggregated Minimum Cost Network Flow Problem

(UAMCNFP)

$$\min \sum_{(i,j)\in A} x_{ij}c_{ij}$$

s.t.

$$\sum_{j:(i,j)\in A} x_{ij} - \sum_{j:(j,i)\in A} x_{ji} = b_i \qquad \forall i \in N$$

$$l_{ij} \le x_{ij} \le u_{ij} \qquad \forall (i,j) \in A$$

Where:

x_{ij} = flow on arc $(i,j) \in A$

c_{ij} = cost for sending one unit flow on arc $(i,j) \in A$

l_{ij} = lower bound on the flow for arc $(i,j) \in A$

u_{ij} = flow capacity of arc $(i,j) \in A$

b_i = supply at node $i \in N$ (negative supply $\hat{=}$ demand)

$N = \{1,...,i,...,m\}$ the set of nodes; $N = S \cup I \cup D$, where S is the set of sources, D the set of destinations and I the set of intermediate nodes

A = the set of arcs, indexed by (i,j) where $i,j \in N$

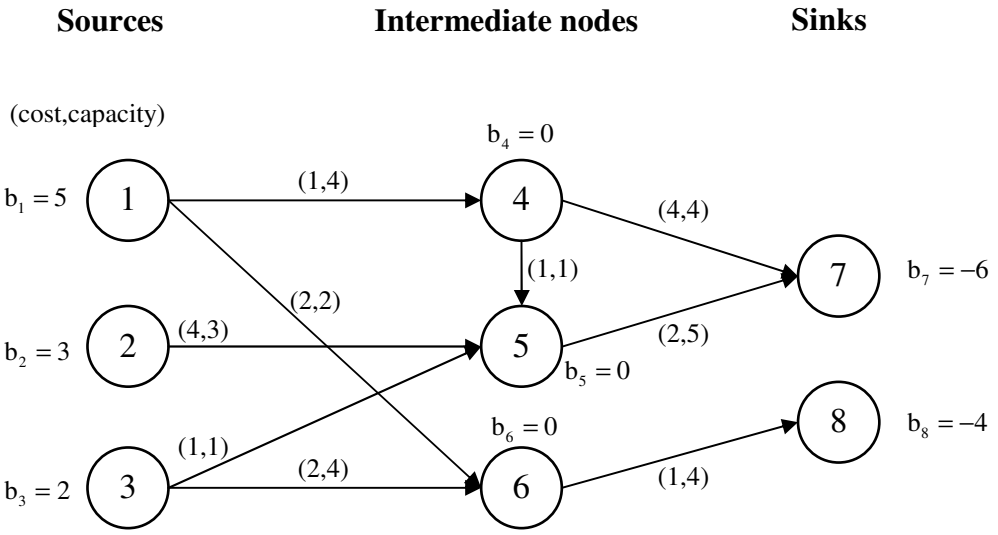
Let us denote an optimal solution by $x^* = (x_{ij}^*)$, with cost z^*. We further assume that $\sum_{i \in S} b_i = -\sum_{i \in D} b_i$

Remark: Unlike the transportation problem it is possible that no feasible solution for the UAMCNFP exists, even though if $\sum_{i \in S} b_i = -\sum_{i \in D} b_i$ holds. Therefore, we assume in the following that a feasible solution for the UAMCFP exists.

Without loss of generality we assume in the following that $l_{ij} = 0 \; \forall (i, j) \in A$ (e.g. replace x_{ij} by $x'_{ij} + l_{ij}$ see for example [AMO93])

The following example gives an overview about the definitions made so far.

Example 5.1



Where $x_{14} = 3;\; x_{16} = 2;\; x_{25} = 3;\; x_{36} = 2;\; x_{47} = 3;\; x_{57} = 3; x_{68} = 4;$ is a feasible solution with objective value $z = 45$.

Due to the structure of the UAMCNFP, it can be solved by special algorithms. Therefore, it seems to be reasonable to preserve its inherent structure. Hence, the final result of aggregation should be again a minimum cost network flow problem.
To summarize this, it can be achieved by grouping nodes of the UAMCFP in one single node and creating arcs among these new nodes. To derive the corresponding parameters, *respecification maps* for the recalculation of costs, supply and demand as well as of capacities have to be applied.
Every aggregation will be based on a partition $\overline{NP}$ of the node set *N*. Therefore, let us have a closer look on this definition, before we come to the formal description of the aggregated minimum cost network flow problem.

Definition 5.1

Let $\overline{NP} = \{J_k : J_k \subseteq N\}$ be a partition of the node set *N* satisfying

i.) $\bigcup_{J_k \in \overline{NP}} J_k = N$

ii.) $J_k \cap J_l = \varnothing \qquad \forall J_k, J_l \in \overline{NP},\ k \neq l$

Example 5.2

A possible partition of the node set of Example 5.1 is given here:

$J_1 = \{1,2\};\ J_2 = \{3\};\ J_3 = \{4,5\};\ J_4 = \{6\};\ J_5 = \{7,8\}$

Given a partition $\overline{NP}$ of the node set *N* of the original problem, the aggregated minimum cost network flow problem can be described as follows:

(AMCNFP)

$$\min \sum_{(n,p)\in \overline{A}} \overline{y}_{np}\,\overline{c}_{np}$$

s.t.

$$\sum_{p:(n,p)\in \overline{A}} \overline{y}_{np} - \sum_{p:(p,n)\in \overline{A}} \overline{y}_{pn} = \overline{b}_n \qquad \forall n \in \overline{N}$$

$$0 \le \overline{y}_{np} \le \overline{u}_{np} \qquad \forall (n,p) \in \overline{A}$$

Where:

$\overline{y}_{np}$ = flow on arc $(n,p) \in \overline{A}$

$\overline{c}_{np}$ = cost for sending one unit flow on arc $(n,p) \in \overline{A}$

$\overline{u}_{np}$ = flow capacity of arc $(n,p) \in \overline{A}$

$\overline{b}_n$ = supply at node $n \in \overline{N}$ (negative supply $\hat{=}$ demand)

$\overline{N} = \{k : J_k \in \overline{NP}\} = \{1,...k,...m_{\overline{NP}}\}$ the set of aggregated nodes; $\overline{N} = \overline{S} \cup \overline{I} \cup \overline{D}$

$\overline{A}$ = the set of aggregated arcs, indexed by (n,p) where $n,p \in \overline{N}$

Any finite optimal solution is denoted by $\overline{y}^* = (\overline{y}^*_{np})$, with cost $\overline{z}^*$.

We can observe in the problem definition above that the nodes combined in the subset J_k are represented through node $k \in \overline{N}$ in the aggregated problem. The corresponding arcs are defined according to the following rule:

There is an arc from node *n* to node *p* where $n, p \in \overline{N}$ and $n \neq p$

$\Leftrightarrow$ there exists at least one arc $(i, j) \in A$ from a node $i \in J_n$ to a node $j \in J_p$ in the original network

There exist different possibilities for defining the costs and capacities of the aggregated arcs. Lee and Francis used the approach of Balas and defined the following respecification maps:

Aggregation by Dominance

$$\bar{c}_{np} = \min_{\substack{i \in J_n, j \in J_p: \\ (i,j) \in A}} c_{ij} \qquad (\bar{c}_{min})$$

$$\bar{u}_{np} = \sum_{\substack{i \in J_n, j \in J_p: \\ (i,j) \in A}} u_{ij} \qquad (\bar{u}_{\Sigma})$$

$$\bar{b}_n = \sum_{i \in J_n} b_i$$

The proposition below shows us that using the respecification maps defined above results in a feasible aggregated problem (if we assume that the original problem has a feasible solution).

Proposition 5.1

If the UAMCNFP has a feasible solution, then the AMCNFP derived by using $\bar{u}_{\Sigma}$ as respecification map for the capacity and $\bar{b}_n$ for the supply/demand, has also a feasible solution, independent from the choice of $\overline{NP}$.

Proof:

Let *x* be a feasible solution for the UAMCNFP.

Define $\bar{y}_{np} = \sum_{\substack{i \in J_n, j \in J_p: \\ (i,j) \in A}} x_{ij} \qquad \forall (n,p) \in \bar{A}$

To show: i.) $\sum_{p:(n,p) \in \bar{A}} \bar{y}_{np} - \sum_{p:(p,n) \in \bar{A}} \bar{y}_{pn} = \bar{b}_n \qquad \forall n \in \bar{N}$

ii.) $0 \le \bar{y}_{np} \le \bar{u}_{np} \qquad \forall (n,p) \in \bar{A}$

To i.)

$$\sum_{p:(n,p) \in \bar{A}} \bar{y}_{np} - \sum_{p:(p,n) \in \bar{A}} \bar{y}_{pn} = \sum_{p:(n,p) \in \bar{A}} \sum_{\substack{i \in J_n, j \in J_p: \\ (i,j) \in A}} x_{ij} - \sum_{p:(p,n) \in \bar{A}} \sum_{\substack{j \in J_p, i \in J_n: \\ (j,i) \in A}} x_{ji}$$

$$= \sum_{i \in J_n} \sum_{p:(n,p) \in \bar{A}} \sum_{j \in J_p:(i,j) \in A} x_{ij} - \sum_{i \in J_n} \sum_{p:(p,n) \in \bar{A}} \sum_{j \in J_p:(j,i) \in A} x_{ji}$$

$$= \sum_{i \in J_n} \underbrace{\left(\sum_{j:(i,j) \in A} x_{ij} - \sum_{j:(j,i) \in A} x_{ji} \right)}_{= b_i} = \sum_{i \in J_n} b_i = \bar{b}_n$$

To ii.)

$$0 \le \bar{y}_{np} = \sum_{\substack{i \in J_n, j \in J_p: \\ (i,j) \in A}} x_{ij} \le \sum_{\substack{i \in J_n, j \in J_p: \\ (i,j) \in A}} u_{ij} = \bar{u}_{np}$$

q.e.d.

Coming to the respecification maps of Zipkin. He used some kind of convex combination to derive the costs for the aggregated problem:

Weighted Aggregation

$$\bar{c}_{np} = \sum_{i \in J_n} \sum_{j \in J_p} g_{ij}^{np} c_{ij} \qquad (\bar{c}_{conv})$$

$$\bar{u}_{np} = \min_{i \in J_n, j \in J_p} \{ \frac{u_{ij}}{g_{ij}^{np}} : \; g_{ij}^{np} > 0 \} \qquad (\bar{u}_{min})$$

$$\bar{b}_n = \sum_{i \in J_n} b_i$$

Where:

$$g_{ij}^{np} = g_i^n g_j^p$$

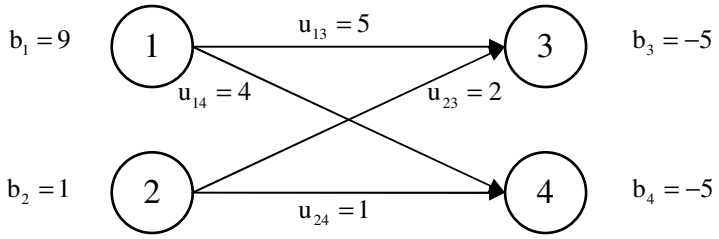
$$g_i^n = \begin{cases} \frac{b_i}{\bar{b}_n}; & \forall i \in S \\ \frac{|b_i|}{\bar{b}_n}; & \forall i \in D \\ \in [0,1] \text{ s.t.} \sum_{i \in J_n} g_i^n = 1 & \forall i \in I \end{cases}$$

Note: When summing over $i \in J_n$ and $j \in J_p$ in $\bar{c}_{conv}$ we do not have to take into account if $(i,j) \in A$ holds. This can be done because particular assumptions stated by Zipkin assure that all nodes grouped together in a subset J_k have the same predecessors and successors, respectively. We refer to Section 5.3 for further details about these assumptions.

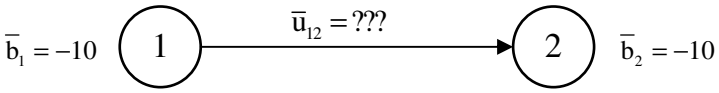
Using the aggregation by dominance approach leads to a feasible aggregated problem as long as the original problem is feasible. Therefore it would be interesting to know if this holds for the weighted aggregation approach, too.
Unfortunately, we see in the following example that even though the original problem is feasible, the aggregated problem derived by using Zipkin's weighted aggregation approach can be infeasible.

Example 5.3

UAMCNFP



AMCNFP



$\bar{u}_{12} = \min\{11,1; 8,9; 40; 20\} = 8,9$

In the same way rounding up to nine does not make the aggregated problem feasible.

Up to now we presented two approaches for the derivation of parameters. Both are different in the definition of costs and capacities and have different advantages and drawbacks. Of course we can also think of other definitions, depending on the current application. You will find such a situation in Chapter 6 about the application of aggregation to the evacuation problem. In it, we will have to define problem dependent respecification maps.

The respecification maps used by Lee and Francis can be calculated easily. Multiplication and division, possibly resulting in non-integer parameters, are not required. It can also be observed that the AMCNFP corresponding to an aggregation by dominance is a relaxation of the UAMCNFP (see Theorem 5.1 in the next section).
Although the computational effort of calculating Zipkin's parameter is much higher and may result in an infeasible AMCNFP, it has the useful property, that the *fixed-weight* disaggregation can directly applied on it, resulting quickly in a feasible (possibly non-integer) solution for the UAMCNFP. In the following sections, we will discuss the two approaches in detail.

Before we finish this section with an aggregation algorithm and a concluding example, let us finally denote,

$$
\begin{aligned}
X &= \text{the set of feasible solutions of an UAMCNFP} \\
\bar{Y} &= \text{the set of feasible solutions of an AMCNFP} \\
\text{pred}(i) &= \{j \in N : (j,i) \in A\} \\
\text{succ}(i) &= \{j \in N : (i,j) \in A\} \\
n(i) &= \text{index } n \in \bar{N} \text{ s.t. } i \in J_n, i \in N \\
p(j) &= \text{index } p \in \bar{N} \text{ s.t. } j \in J_p, j \in N \\
(\bar{\pi}, \bar{\alpha}) &= (\bar{\pi}_n, \bar{\alpha}_{np}) = \text{an optimal solution to the dual of the AMCNFP}
\end{aligned}
$$

The following algorithm enables us to build up an aggregated minimum cost network flow problem based on an original problem and a corresponding node partition $\overline{NP}$.

Aggregation Algorithm for the Minimum Cost Network Flow Problem

INPUT: UAMCNFP, a partition $\overline{NP}$ of the corresponding node set

OUTPUT: An aggregated minimum cost network flow problem according to the partition $\overline{NP}$ of the node set of the original problem

STEP 1: Set $\overline{\mathrm{N}} = \varnothing; \overline{\mathrm{A}} = \varnothing$

STEP 2: Each subset of original nodes, J_k, is replaced through a single aggregated node specified with *k*.
$\overline{\mathrm{N}} = \{\mathrm{k} : \mathrm{J_k} \in \overline{\mathrm{NP}}\} = \{1,...,\mathrm{k},...\mathrm{m}_{\overline{\mathrm{NP}}}\}$
Determine the supply/demand for each aggregate node

STEP 3: For each node $n \in \overline{N}$ Do
For each node $p \in \overline{N}, p \neq n$ Do
If there exits at least one arc $(i, j) \in A$ for $i \in J_n$ and $j \in J_p$ then
$\overline{\mathrm{A}} = \overline{\mathrm{A}} \cup (\mathrm{n}, \mathrm{p})$

STEP 4: For each arc $(n, p) \in \overline{A}$ determine the corresponding costs and capacities.

Remark: 1.) The algorithm requires a partition $\overline{NP}$ of the node set at the beginning. In Section 5.4 we will have a look on how a reasonable partition can be derived. However, we can already say that in most cases the selection of $\overline{NP}$ is problem dependent. This means that the partition is determined subjectively and logically for the particular problem.

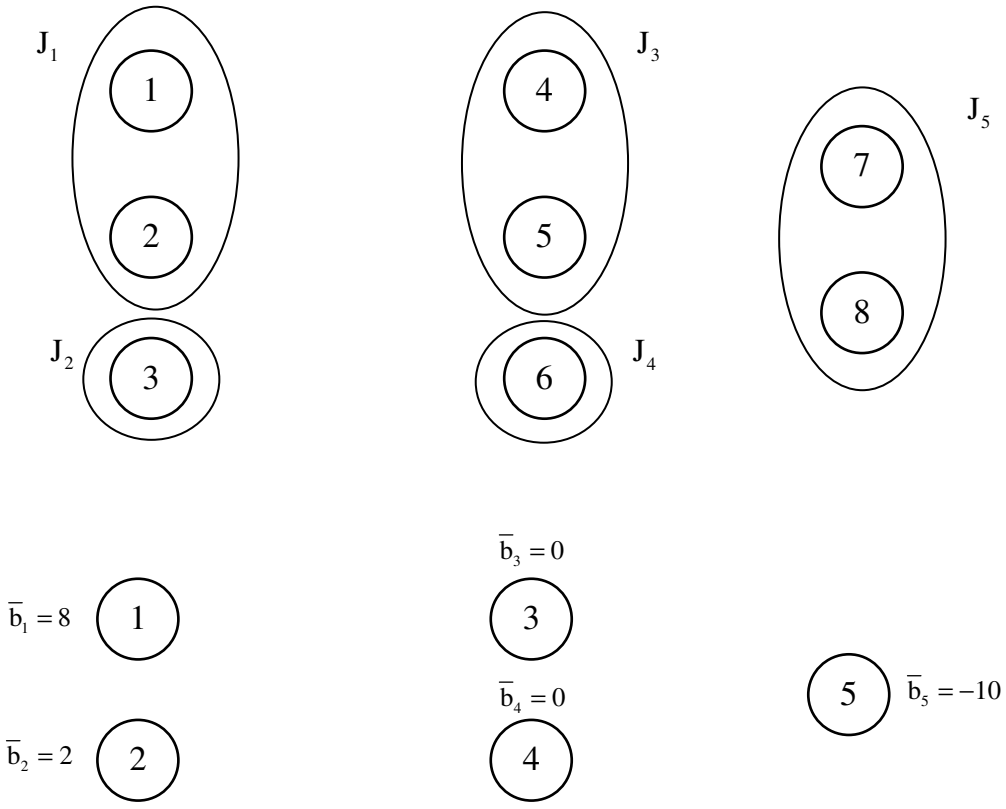
2.) For calculating the parameters of the aggregated arcs, the presented approaches can be used.

We finish this section with an example showing the proceeding of the algorithm and how an aggregated problem finally looks like.

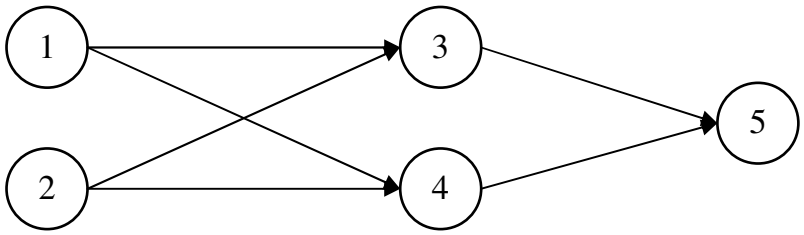
Example 5.4

Taking $\overline{NP} = \{J_1, J_2, J_3, J_4, J_5\}$ from Example 5.2 and the MCNFP of Example 5.1, we get for the different steps of the algorithm:

STEP 2:



STEP 3:



STEP 4:

The following table gives an overview about the costs and capacities of the aggregated problem, where we used the aggregation by dominance approach for calculating the parameters:

From/To (cost , cap)	1	2	3	4	5
1	**-**	-	(1,7)	(2,2)	-
2	-	-	(1,1)	(2,4)	-
3	-	-	-	-	(2,9)
4	-	-	-	-	(1,4)
5	-	-	-	-	-

Table VIII: Cost and Capacity for the aggregated problem corresponding to the original one of Example 5.1

We have not used the respecification maps of Zipkin, because we will see that the UAMCNFP of the example above does not fit the assumptions on aggregation formulated by Zipkin. We refer to Example 5.8, in which we have an original network satisfying the assumptions.
As we have already seen for the transportation problem, aggregation causes a loss of accuracy. This loss of accuracy mainly depends on the respecification map used for defining the parameters of the aggregated problem. Therefore, we will examine in the following, besides the approaches of Lee/Francis and Zipkin for solving large-scale minimum cost network flow problems, also the loss of accuracy induced by applying their respecification maps.

5.2 Aggregation by Dominance

Recall:

$$\overline{c}_{np} = \min_{\substack{i\in J_n, j\in J_p:\\ (i,j)\in A}} c_{ij} \qquad (\overline{c}_{min})$$

$$\overline{u}_{np} = \sum_{\substack{i\in J_n, j\in J_p:\\ (i,j)\in A}} u_{ij} \qquad (\overline{u}_{\Sigma})$$

$$\overline{b}_n = \sum_{i\in J_n} b_i$$

The approaches of Lee and Francis are based on the fundamental ideas of Balas about the solution of large-scale transportation problems. They took up his concepts and developed an algorithm for large-scale minimum cost network flow problems. Lee provided in his Ph. D. thesis [Lee75] the basic concepts for such an algorithm including a termination criterion which detects optimality. The different steps of the algorithm are described in a very general way and leave several questions open. Francis took up this general version of the algorithm and included very detailed theory for each step of the algorithm.
The basic idea of the algorithm can be summarized as follows: starting with the aggregation of the UAMCNFP based on a partition $\overline{NP}^r$ and solve the corresponding AMCNFP^r to optimality. A new aggregated problem AMCNFP^{r+1} is derived by refining a subset of $\overline{NP}^r$ into two new subsets. The disaggregation of the solution of the AMCNFP^r serves as start solution for the AMCNFP^{r+1}. These steps are repeated until the termination criterion is satisfied, which detects an optimal solution.
The termination criterion is a central point in Lee's as well as in Francis' concepts. For our formulation of the MCNFP we will have to make some slide changes on this criterion.
Before we discuss the algorithm, we will present an interesting characteristic of the aggregation by dominance approach, which is applied for computing the parameters of the aggregated problem. The section will be concluded with a bound for the error caused by aggregation, when the algorithm stops before an optimal solution is detected and a discussion of the advantages and drawbacks of the presented approach.

The following theorem indicates that an aggregated minimum cost network flow problem derived by using $\overline{c}_{min}$ and $\overline{u}_{\Sigma}$ as respecification maps results in a relaxation of the original problem.

Theorem 5.1

Given an UAMCNFP and a corresponding AMCNFP based on a partition of the node set $\overline{NP}$ and derived by using the aggregation by dominance approach. Then it holds that the aggregated problem is a relaxation of the original one.

Proof:

To show: i.) $X \subseteq \overline{Y}$

ii.) $$\overline{z}(x) = \sum_{(n,p)\in\overline{A}} \overline{c}_{np} \sum_{\substack{i\in J_n, j\in J_p: \\ (i,j)\in A}} x_{ij} \leq \sum_{(i,j)\in A} c_{ij}x_{ij} = z(x) \quad \forall x \in X$$

To i.) See proof of Proposition 5.1

To ii.)

$$\overline{z}(x) = \sum_{(n,p)\in\overline{A}} \overline{c}_{np} \sum_{\substack{i\in J_n, j\in J_p: \\ (i,j)\in A}} x_{ij} = \sum_{(n,p)\in\overline{A}} \min_{\substack{i\in J_n, j\in J_p: \\ (i,j)\in A}} c_{ij} \sum_{\substack{i\in J_n, j\in J_p: \\ (i,j)\in A}} x_{ij}$$

$$\leq \sum_{(n,p)\in\overline{A}} \sum_{\substack{i\in J_n, j\in J_p: \\ (i,j)\in A}} c_{ij}x_{ij} = \sum_{(i,j)\in A} c_{ij}x_{ij} = z(x)$$

q.e.d.

Corollary 5.1

Given an UAMCNFP and a corresponding AMCNFP based on a partition of the node set $\overline{NP}$ and derived by using the aggregation by dominance approach. Then the following inequality holds:

$$\overline{z}^* \leq z^*$$

Neither Lee nor Francis derived a bound for this quantity, which provides information how far we are away from optimality at most with our solution of the relaxed problem. At the end of this section we will present such a bound.

5.2.1 Lee's and Francis' Algorithm for Large-Scale MCNFP

We saw so far that the aggregated problem using $\overline{c}_{min}$ and $\overline{u}_{\Sigma}$ as respecification maps, results in a relaxation. However, the approach of Lee and Francis finally leads to an optimal solution. The starting point of their algorithm is the solution of an AMCNFP. By means of stepwise refinement, a sequence of aggregated problems is derived. The refinement process stops if an optimal solution, characterized by a termination criterion, has been found.
Before we go on with the general proceeding of the algorithm, let us have a more formal look what refinement in that case means. If we assume that we are in iteration *r* of the algorithm we have the following situation:

Situation before Refinement

$$\overline{\mathrm{NP}}^{\,\mathrm{r}} = \{\mathrm{J}_1^{\mathrm{r}},...,\mathrm{J}_{\mathrm{k}}^{\mathrm{r}},...,\mathrm{J}_{\mathrm{m}}^{\mathrm{r}}\}$$
$$\overline{\mathrm{N}}^{\,\mathrm{r}} = \{1,...,\mathrm{k},...\mathrm{m}_{\mathrm{r}}\}$$

In order to get a partition of the node set for iteration $r+1$, we divide a subset J_k^r into two new subsets. The following definition describes this proceeding in a more formal way.

Definition 5.2

Let $\overline{NP}^{\,r}$ be a partition of the node set of an UAMCNFP.
Refining subset J_k^r into two new subsets results in the new partition $\overline{NP}^{\,r+1}$ defined as follows:

$$\overline{\mathrm{NP}}^{\,\mathrm{r+1}} = \{\mathrm{J}_1^{\mathrm{r+1}},...,\mathrm{J}_{\mathrm{k}}^{\mathrm{r+1}},...,\mathrm{J}_{\mathrm{m}}^{\mathrm{r+1}},\mathrm{J}_{\mathrm{m+1}}^{\mathrm{r+1}}\}$$
$$\overline{\mathrm{N}}^{\,\mathrm{r+1}} = \{1,...,\mathrm{k},...,\mathrm{m}_{\mathrm{r}},\mathrm{m}_{\mathrm{r+1}}\}$$

Where:

$$\mathrm{J}_{\mathrm{q}}^{\mathrm{r+1}} = \mathrm{J}_{\mathrm{q}}^{\mathrm{r}} \qquad \forall \mathrm{k} \neq \mathrm{q} \neq \mathrm{m}+1$$
$$\mathrm{J}_{\mathrm{m+1}}^{\mathrm{r+1}} \subset \mathrm{J}_{\mathrm{k}}^{\mathrm{r}}$$
$$\mathrm{J}_{\mathrm{k}}^{\mathrm{r+1}} = \mathrm{J}_{\mathrm{k}}^{\mathrm{r}} \cap \mathrm{J}_{\mathrm{m+1}}^{\mathrm{r+1}}$$
$$\mathrm{J}_{\mathrm{k}}^{\mathrm{r+1}} \cap \mathrm{J}_{\mathrm{m+1}}^{\mathrm{r+1}} = \varnothing$$

The definition above is a simple refinement of subset J_k^r. By iterative application of simple refinements more general refinements can be achieved. The following example shows the refinement process.

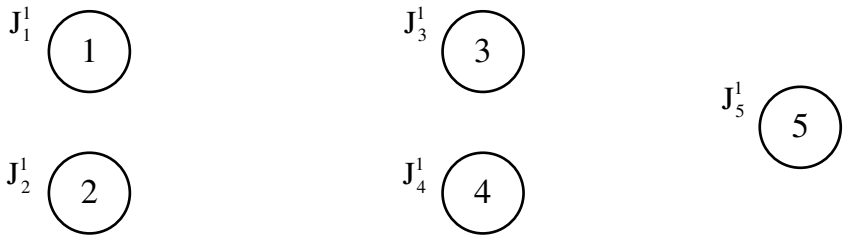
Example 5.5

Taken the partition of the node set of Example 5.1, where we have the following initial situation:

$\overline{\mathrm{NP}}^1 = \{\mathrm{J}_1^1, \mathrm{J}_2^1, \mathrm{J}_3^1, \mathrm{J}_4^1, \mathrm{J}_5^1\}$

$\overline{\mathrm{N}}^1 = \{1, 2, 3, 4, 5\}$

Graphical Representation



Refining J_3^1 *yields the following new partition of the node set*

$\overline{\mathrm{NP}}^2 = \{\mathrm{J}_1^2, \mathrm{J}_2^2, \mathrm{J}_3^2, \mathrm{J}_4^2, \mathrm{J}_5^2, \mathrm{J}_6^2\}$

$\overline{\mathrm{N}}^2 = \{1, 2, 3, 4, 5, 6\}$

Where

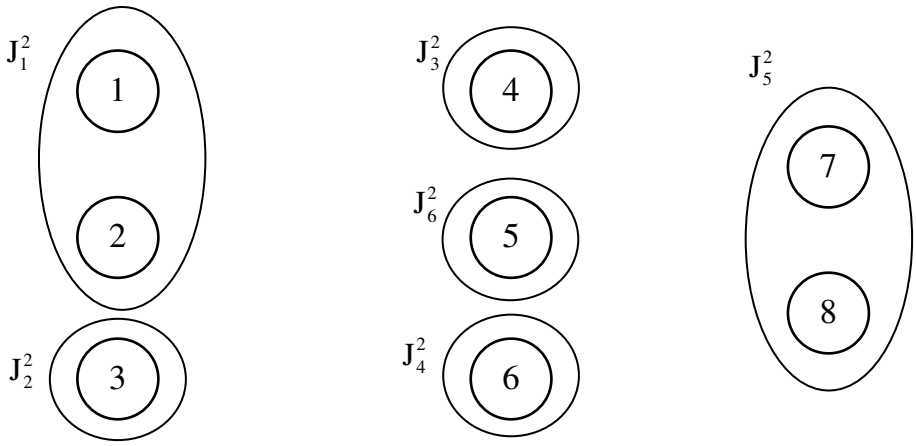
$\mathrm{J}_1^2 = \mathrm{J}_1^1;\ \mathrm{J}_2^2 = \mathrm{J}_2^1;\ \mathrm{J}_4^2 = \mathrm{J}_4^1;\ \mathrm{J}_5^2 = \mathrm{J}_5^1;$

$\mathrm{J}_3^2 = \{4\};$

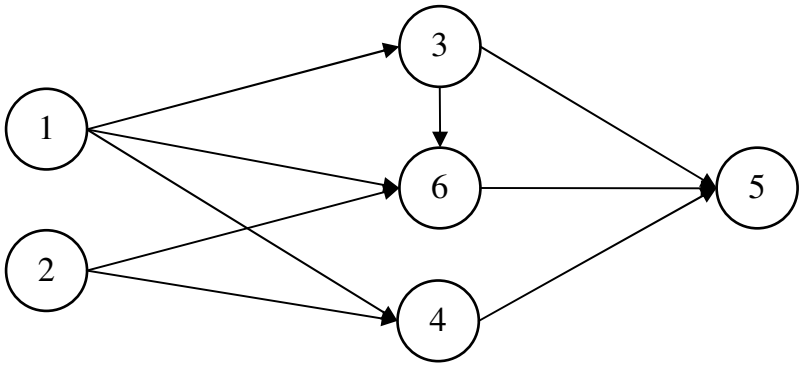
$\mathrm{J}_6^2 = \{5\};$

Graphical Representation

Partition of the original problem based on $\overline{NP}^2$



Corresponding aggregated problem based on $\overline{N}^2$



Now we are able to present the algorithm, which is based on the ideas of Lee. We also have added some parts of Francis' concepts

Algorithm for solving Large-Scale Minimum Cost Network Flow Problems by Aggregation

INPUT: UAMCNFP, an initial partition $\overline{NP}^1$ of the node set *N*, a corresponding AMCNFP[1]

OUTPUT: An optimal solution to the UAMCNFP

STEP 1: Set r = 1;
Solve AMCNFP[r] to optimality. Let $\overline{y}_r^*$ an optimal solution for this problem

STEP 2: Make a simple refinement (e.g. a refinement into two new subsets) of the subset J_k^r and let $\overline{NP}^{r+1}$ the corresponding new partition of the node set

STEP 3: Derive AMCNFP[r+1] based on $\overline{NP}^{r+1}$

STEP 4: Use the solution $\overline{y}_r^*$ of AMCNFP[r] to get a starting solution for AMCNFP[r+1] by setting:

$$(\overline{y}_{np})_{r+1} = (\overline{y}_{np}^*)_r \qquad \forall (n,p) \in \overline{A} \text{ where } n \neq k \neq p, n \neq m_{r+1} \neq p$$

On arcs that have *k* or/and m_{r+1} as tail or head node the flow $(\overline{y}_{nk}^*)_r$ and $(\overline{y}_{kp}^*)_r$ has to be split up satisfying the following network flow problem:

$$\sum_{p:(k,p)\in \overline{A}^{r+1}} (\overline{y}_{kp})_{r+1} - \sum_{p:(p,k)\in \overline{A}^{r+1}} (\overline{y}_{pk})_{r+1} = \overline{b}_k^{r+1}$$

$$\sum_{p:(m_{r+1},p)\in \overline{A}^{r+1}} (\overline{y}_{m_{r+1}p})_{r+1} - \sum_{p:(p,m_{r+1})\in \overline{A}^{r+1}} (\overline{y}_{pm_{r+1}})_{r+1} = \overline{b}_{m_{r+1}}^{r+1}$$

$$(\overline{y}_{np})_{r+1} \le \overline{u}_{np}^{r+1} \qquad \forall (n,p)\in \overline{A}^{r+1} : n\in\{k, m_{r+1}\} \text{ or / and } p\in\{k, m_{r+1}\}$$

$$(\overline{y}_{kp})_{r+1} + (\overline{y}_{m_{r+1}p})_{r+1} = (\overline{y}^*_{kp})_r \qquad \forall\, p\in \overline{N}^{r+1} : (m_{r+1}, p) \text{ or / and } (k,p)\in \overline{A}^{r+1};$$

$$(\overline{y}_{nk})_{r+1} + (\overline{y}_{nm_{r+1}})_{r+1} = (\overline{y}_{nk})_r \qquad \forall\, n\in \overline{N}^{r+1} : (n, m_{r+1}) \text{ or / and } (n,k)\in \overline{A}^{r+1};$$

STEP 5: Solve $AMCNFP^{r+1}$ to optimality.

STEP 6: If the *termination criterion* is fulfilled

→ STOP $\overline{y}^*_{r+1}$ is an optimal solution for UAMCNFP

Else

→ Set r = r + 1 and GOTO STEP 2

Remark:

1.) Because of the fact that the AMCNFP is a relaxation of the original problem, there might be no feasible solution for the problem in STEP 4 (e.g. violation of the flow constraints in node k respective m_{r+1}). An artificial arc between k and m_{r+1} or in the opposite direction can be added to get a feasible solution. By assigning high cost to the artificial arc it will be priced out if a true feasible solution exists.

2.) There is no statement how a "good" refinement of $\overline{NP}^r$ in STEP 2 should look like. Francis [Fra85] made some further evaluations regarding this topic.

Before we go on with some open questions regarding the different steps of the algorithm, we should state the missing termination criterion. We had to make some slide changes, because in contrast to Lee we do not have a circulation problem.

Termination Criterion [Lee75] (Updated Version)

Given an UAMCNFP and a partition of the node set $\overline{NP}$. Let $\overline{y}^*$ be an optimal solution to the AMCNFP. If we assume that J_n consists entirely of sources, destinations or intermediate nodes $\forall J_n \in \overline{NP}$ and further all non-zero values of $\overline{y}^*$ correspond to arcs $(n,p)\in \overline{A}$ generated from singleton sets $J_n = \{i\}$ and $J_p = \{j\}$, then the refinement process can be terminated.

The following theorem is based on the ideas of Lee.

Theorem 5.2:

Let $\bar{y}^*$ be an optimal solution to AMCNFP satisfying the termination criterion in the updated version. Define a solution *x* to the UAMCNFP as follows:

$$x_{ij} = \begin{cases} \bar{y}^*_{np}, \text{ if } J_n = \{i\} \text{ and } J_p = \{j\} \\ 0, \text{ else} \end{cases}$$

Then *x* is an optimal solution to UAMCNFP with corresponding objective value $\bar{z}^*$.

Proof:

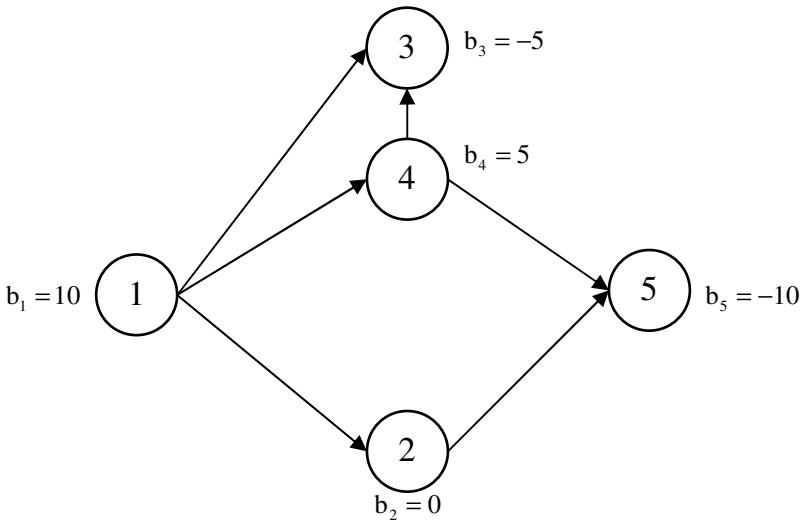
There is no flow on arcs $(n, p) \in \bar{A}$, if $|J_n|$ or/and $|J_p|$ is greater than one. This means that all non-zero flow is on arcs and between nodes corresponding to the original problem (UAMCNFP). $\bar{y}^*$ is feasible for the AMCNFP, hence the flow conservation and capacity constraints are satisfied. Because of the fact that the flow is only send on parts belonging to the original problem, *x* as defined above is also feasible for UAMCNFP. Since we know from Corollary 5.1 that $\bar{z}^* \leq z^*$, *x* is an optimal solution for UAMCNFP.

q.e.d.

Remark: The assumption that only nodes of the same type are grouped together is required to avoid pseudo transshipment nodes. See the following example.

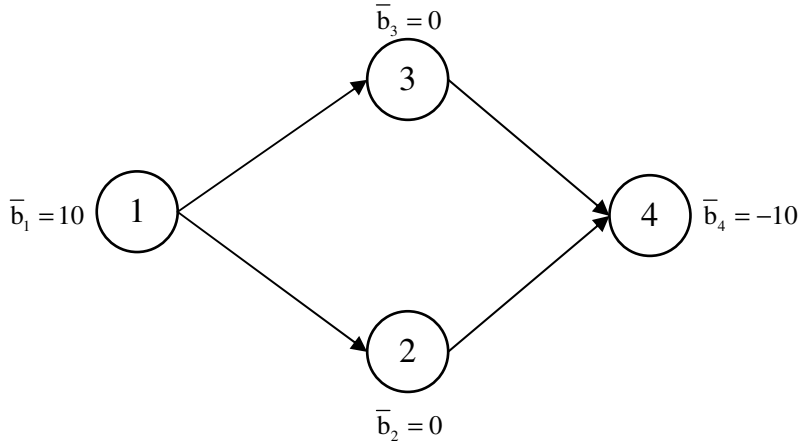
Example 5.6

UAMCFP



By grouping node 3 of 4 of the original network, we get the following aggregated problem:

AMCNFP



Sending a flow of value ten on the path 1→ 2→ 4 yields an optimal solution for the aggregated problem, where the costs and capacities of the original problem are defined appropriate. The original version of the termination criteria of Lee would stop at this point of the algorithm, because all the non-zero flow is on arcs corresponding to original nodes. However, the solution of the AMCNFP is not feasible for the original problem.

Back to the remaining steps of the algorithm. Because we only presented the basic ideas of the algorithm, some questions remain open.

- In STEP 2 a refinement of $\overline{NP}^r$ has to be chosen, how should this refinement look like?
- In STEP 4 the flow of solution $\bar{y}_r^*$ is split up to get a solution for the aggregated problem AMCNFP^{r+1}, how should the disaggregation map be defined in order to get a basic start solution for AMCNFP^{r+1}?

In the Ph. D. Thesis of Francis [Fra85] these open questions are addressed. Therefore, let us have a brief look on his extensions.
In iteration *r* we have to solve a network sub problem $\overline{NSP}^r$ in order to get a starting solution for the aggregated problem in iteration $r+1$. If the $\overline{NSP}^r$ is kept small, then the advanced-start basic solution for problem $r+1$ can be obtained very quickly with less computational effort. Therefore, Francis took the size of $\overline{NSP}^{r+1}$ as a measure to decide on how the refinement of $\overline{NP}^r$ into $\overline{NP}^{r+1}$ should look like. He derived a heuristic resulting in a small $\overline{NSP}^{r+1}$, which seems to be a valuable criterion for any network optimization code using aggregation-disaggregation concepts. The derived heuristic can be used in STEP 2 to get a reasonable refinement. He also developed a disaggregation map which transforms a *basic* feasible (optimal) solution of problem *r* into a *basic* solution for problem $r+1$. The advantage

of such a mapping strategy is obvious. By disaggregating the solution of $AMCNFP^r$, we get an advanced basic starting solution for $AMCNFP^{r+1}$. The information available from the optimal solution of problem *r* would thus be effectively used to reduce the computational effort required to solve problem $r+1$. For an efficient network algorithm which uses aggregation-disaggregation methods, it is important to have a disaggregation map with such a property. Parts of this diasaggregation map are already included in the previous algorithm. Francis however focused more on this mapping process and suits this procedure to the algorithm, a primal-dual algorithm, he used to solve the different minimum cost network flow problems. For more details of Francis' enhancements we refer his dissertation mentioned above.

5.2.2 Bound on the Loss of Accuracy

We have seen so far how we can get an optimal solution for large MCNFP by solving a sequence of aggregated problems. Sometimes, however, it is enough to have a proper solution for the problem, which is close enough to optimality. Therefore, it would be useful to have a bound, which provides us with the information how far we are away from optimality in the worst case. By taking up basic duality theory we can derive a bound for the case we use $\overline{c}_{min}$ and $\overline{u}_{\Sigma}$ as respecification maps.

Proposition 5.2

Given an original minimum cost network flow problem and a corresponding aggregated problem based on a partition $\overline{NP}$ of the node set and derived by using the aggregation by dominance approach. Then the following inequality holds:

$$z^* - \overline{z}^* \le \max_{x \in X} \sum_{(i,j)\in A} [c_{ij} - (\overline{\pi}_{n(i)} - \overline{\pi}_{p(j)})]\, x_{ij} + \sum_{(n,p)\in \overline{A}} \overline{\alpha}_{np}\, \overline{u}_{np}$$

Proof:

$$z^* = \sum_{(i,j)\in A} c_{ij} x^*_{ij} = \sum_{(i,j)\in A} c_{ij} x^*_{ij} + \sum_{i\in N} \overline{\pi}_{n(i)} \underbrace{[b_i - (\sum_{j:(i,j)\in A} x^*_{ij} - \sum_{j:(j,i)\in A} x^*_{ji})]}_{=0}$$

$$= \sum_{(i,j)\in A} c_{ij} x^*_{ij} + \sum_{i\in N} \overline{\pi}_{n(i)} [b_i - (\sum_{j:(i,j)\in A} x^*_{ij} - \sum_{j:(j,i)\in A} x^*_{ji})] + \sum_{(n,p)\in \overline{A}} \overline{\alpha}_{np} \underbrace{(\overline{u}_{np} - \overline{u}_{np})}_{=0}$$

$$= \underbrace{\sum_{n\in \overline{N}} \overline{\pi}_n \overline{b}_n - \sum_{(n,p)\in \overline{A}} \overline{\alpha}_{np} \overline{u}_{np}}_{=\overline{z}^*} + \sum_{(i,j)\in A} c_{ij} x^*_{ij} + \sum_{i\in N} \overline{\pi}_{n(i)} (- \sum_{j:(i,j)\in A} x^*_{ij} + \sum_{j:(j,i)\in A} x^*_{ji}) + \sum_{(n,p)\in \overline{A}} \overline{\alpha}_{np} \overline{u}_{np}$$

$$= \overline{z}^* + \sum_{(i,j)\in A} c_{ij} x^*_{ij} - \sum_{(i,j)\in A} (\overline{\pi}_{n(i)} - \overline{\pi}_{p(j)}) x^*_{ij} + \sum_{(n,p)\in \overline{A}} \overline{\alpha}_{np} \overline{u}_{np}$$

$$= \overline{z}^* + \sum_{(i,j)\in A} [c_{ij} - (\overline{\pi}_{n(i)} - \overline{\pi}_{p(j)})]\, x^*_{ij} + \sum_{(n,p)\in \overline{A}} \overline{\alpha}_{np} \overline{u}_{np}$$

$$\Rightarrow \qquad z^* - \bar{z}^* = \sum_{(i,j)\in A} [c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)})]\, x^*_{ij} + \sum_{(n,p)\in \bar{A}} \bar{\alpha}_{np} \bar{u}_{np}$$

$$\Rightarrow \qquad z^* - \bar{z}^* \le \max_{x\in X} \sum_{(i,j)\in A} [c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)})]\, x_{ij} + \sum_{(n,p)\in \bar{A}} \bar{\alpha}_{np} \bar{u}_{np}$$

q.e.d.

Remark: The derived bound can be seen as an alternative termination criterion in STEP 6 of the algorithm if only an approximate solution is required, i.e. if the objective value in iteration *r* is good enough for the application, the algorithm can stop.

In order to calculate the bound derived above, it would be necessary to solve a problem of the same size as UAMCNFP. Therefore, as for the transportation problem, it makes sense to relax this problem. By dropping all constraints belonging to supply nodes and all capacity constraints, we get the following a posterior bound.

A posteriori bound on the loss of accuracy for the aggregation by dominance approach

$$\bar{z}^* - z^* \le \sum_{d\in D} |b_d|\, LP_d + CAP \qquad (5.1)$$

LP_d denotes the length of the longest path from any source to destination $d \in D$, with $LE_{ij} = c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)})$ defined as the length of arc $(i, j) \in A$ and $CAP = \sum_{(n,p)\in \bar{A}} \bar{\alpha}_{np} \bar{u}_{np}$.

CAP can be calculated easily because $\bar{u}_{np}$ is known in advance and $\bar{\alpha}_{np}$ can be derived by solving AMCNFP. It requires more effort to calculate the longest path LP_d from any source to destination $d \in D$.

Remark:

1.) In order to compute a meaningful bound, it is important that we found no cycle with a positive weight in the network

2.) As for the transportation problem a second a posteriori bound can be obtained by dropping all demand constraints as well as the capacity constraints; the computation of this bound requires the longest path from each source to any destination.

5.2.3 Advantages and Drawbacks of the Aggregation by Dominance Approach

We close this section with some words about the advantages and drawbacks of the presented approach.

Advantages:

- Easy calculation of the parameters for the aggregated problem
- If the original problem is feasible, the aggregated problem has also an feasible solution; independent of the choice of $\overline{NP}$
- Using the defined respecification maps yields a lower bound on the optimal objective value of the original problem (i.e. $\overline{z}^* \leq z^*$)
- The presented algorithm yields an optimal solution for the original problem at the end
- Using Francis' disaggregation map makes the algorithm efficient, since the results of previous calculations are used to get a start solution for problem $r+1$
- If an optimal solution is not required, then the derived bound on the loss of accuracy can also be used as a termination criteria for the algorithm

Drawbacks:

- All the data of the original problem have to be stored until an optimal solution is found
- In the worst case the algorithm leads to an iteration, in which we have to solve the original problem
- The objective values, corresponding to the entries of the sequence of sub problems generated through the algorithm, are not monotonic increasing
- No direct and easy disaggregation mapping

The usefulness of the presented concepts as well as the algorithm mainly depends on the application. If an optimal solution is required for a large-scale minimum cost network flow problem, then the algorithm is very valuable. In case you require an appropriate aggregation, the use of $\overline{c}_{min}$ as respecification map for the costs has to be taken out very carefully. We will come back to this issue in the chapter about the application to a real world problem.

5.3 The Weighted Aggregation Approach of Zipkin

In the following section we will consider the concepts of Zipkin based on the weighted aggregation. Therefore, all aggregated problems are based on the following respecification maps in this section.

Recall:

$$\bar{c}_{np} = \sum_{i \in J_n} \sum_{j \in J_p} g_{ij}^{np} c_{ij} \qquad (\bar{c}_{conv})$$

$$\bar{u}_{np} = \min_{i \in J_n, j \in J_p} \{ \frac{u_{ij}}{g_{ij}^{np}} : g_{ij}^{np} > 0 \} \qquad (\bar{u}_{min})$$

$$\bar{b}_n = \sum_{i \in J_n} b_i$$

Where:

$$g_{ij}^{np} = g_i^n g_j^p$$

$$g_i^n = \begin{cases} \frac{b_i}{\bar{b}_n}; & \forall i \in S \\ \frac{|b_i|}{\bar{b}_n}; & \forall i \in D \\ g_i^n \in [0,1] \quad \text{s.t.} \sum_{i \in J_n} g_i^n = 1 & \forall i \in I \end{cases}$$

Zipkin extended his approach for the transportation problem, the derivation of duality based bounds for the loss of accuracy, to the more general setting of minimum cost network flow problems. In contrast to the transportation problem only two a posteriori bounds for the minimum cost network flow problem can be given. The *fixed-weight* disaggregation presented for the transportation problem can also be applied to the MCNFP, but not without making some assumptions regarding the partition of the node set $\overline{NP}$. Even though Zipkin presented some ideas to satisfy these assumptions, we will see that they are very restrictive, especially for the application to real world problems. In the following, we will consider these assumptions and give an a posterior bound for the loss of accuracy. The section is concluded with a discussion about the advantages and drawbacks of Zipkin's approach.

5.3.1 Assumptions Concerning the Weighted Aggregation

Let us assume that we have a UAMCNFP and a corresponding partition $\overline{NP}$ of the node set. The following assumptions must hold to derive an aggregated problem satisfying the requirements of Zipkin.

(AZ 1) $\forall J_n \in \overline{NP} : J_n$ consists entirely of sources, destinations or intermediate nodes

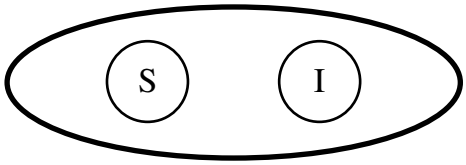
(AZ 2) $\forall J_n \in \overline{NP} : \nexists (i, j) \in A$ s.t. $i, j \in J_n$
(i.e. there are no arcs between nodes in a single subset)

(AZ 3) $\forall J_n \in \overline{NP}$: $\text{pred}(i) = \text{pred}(j)$ and $\text{succ}(i) = \text{succ}(j)$ $\forall i, j \in J_n$
(i.e. all nodes in a subset $J_n \in \overline{NP}$ have identical connections to nodes outside the subset)

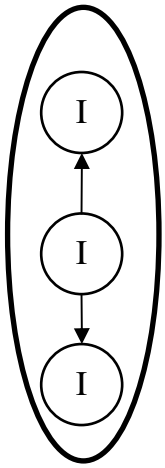
Note: We will assume in the following that the assumptions above are satisfied.

The following example shows the different assumptions of Zipkin.

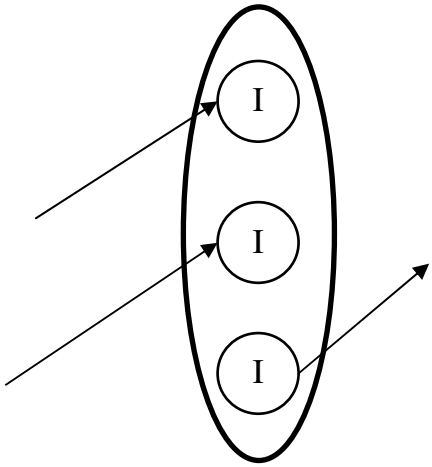
Example 5.7



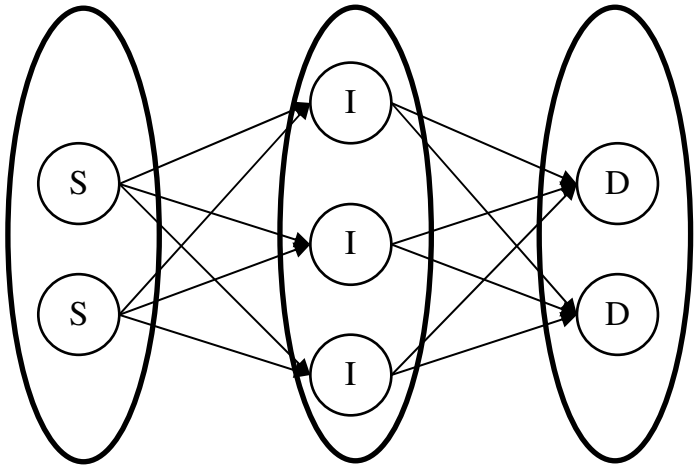
Assumption (AZ 1) fails, since nodes from two types are grouped together



Because of the fact that there are arcs in the original problem between nodes, which are grouped together, assumption (AZ 2) fails



Assumption (AZ 3) fails, since the grouped nodes do not have the same successors and predecessors in the original network



In the latter case all assumption are satisfied.

From our point of view, assumption (AZ 1) is a reasonable one, as we have seen in the last section (see updated version of Lee's termination criterion). Indeed, assumptions (AZ 2) and (AZ 3) may be very restrictive in practice. It is easy to think of problems, in which natural aggregation violates one or both of these assumptions. An example would be the aggregation of transshipment nodes which are geographically and topologically adjacent. Zipkin proposed some possibilities to achieve a simplification of networks in order to satisfy the assumptions. This simplification results in a MCNFP which is either a restriction of the original problem or it is equivalent to the original problem. The application of these approaches seems not to be promising. Therefore, we will not present them in our work.

As we have seen in Section 5.2, the application of $\overline{c}_{min}$ and $\overline{u}_{\Sigma}$ as respecification maps makes the *disaggregation–process* somehow complicated and could result in an infeasible solution for the UAMCFP. Having an AMCNFP based on the weighted aggregation approach and satisfying assumption (AZ 1) – (AZ 3), it is possible to recover a feasible solution for the original problem without much effort by applying the *fixed-weight* disaggregation. The definition of the fixed-weight disaggregation is nearly the same as in the case of the transportation problem and can be formulated as follows.

Definition 5.3

A solution $\overline{x}$ to the UAMCNFP is called a *fixed-weight* solution, if it is derived from a solution $\overline{y}$ of a corresponding AMCNFP in the following way:

$$\overline{x}_{ij} = g_{ij}^{np}\overline{y}_{np} \quad \text{where } i \in J_n, j \in J_p, (n,p) \in \overline{A}$$

Note: In most cases, the fixed-weight solution is not integer.

As in the case of the transportation problem, the fixed-weight disaggregation provides a feasible solution to the UAMCNFP, which will be shown in the following proposition.

Proposition 5.3

Given an original minimum cost network flow problem and a corresponding aggregated problem based on a partition $\overline{NP}$ of the node set.

a.) Let $\overline{y}$ be a feasible solution for the AMCNFP and $\overline{x}$ the corresponding *fixed-weight* solution for the UAMCNFP

$$\Rightarrow \overline{x} \text{ is a feasible solution for the UAMCNFP}$$

b) Let $\overline{y}^*$ be an optimal solution for AMCNFP and $\overline{x}^*$ the corresponding *fixed-weight* solution for the UAMCNFP

$$\Rightarrow \sum_{(i,j)\in A} c_{ij}\overline{x}_{ij}^* = \overline{z}^*$$

Proof:

a.)

To show: i.) $\sum_{j:(i,j)\in A} \overline{x}_{ij} - \sum_{j:(j,i)\in A} \overline{x}_{ji} = b_i \qquad \forall i \in N$

ii.) $0 \le \overline{x}_{ij} \le u_{ij} \qquad \forall (i,j) \in A$

To i.)

$$\sum_{j:(i,j)\in A} \overline{x}_{ij} - \sum_{j:(j,i)\in A} \overline{x}_{ji} = \sum_{j:(i,j)\in A} g_{ij}^{n(i)p(j)}\overline{y}_{n(i)p(j)} - \sum_{j:(j,i)\in A} g_{ji}^{p(j)n(i)}\overline{y}_{p(j)n(i)}$$

$$= \sum_{j:(i,j)\in A} g_i^{n(i)} g_j^{p(j)}\overline{y}_{n(i)p(j)} - \sum_{j:(j,i)\in A} g_j^{p(j)} g_i^{n(i)}\overline{y}_{p(j)n(i)}$$

$$= g_i^{n(i)} \Big(\sum_{j:(i,j)\in A} g_j^{p(j)} \overline{y}_{n(i)p(j)} - \sum_{j:(j,i)\in A} g_j^{p(j)} \overline{y}_{p(j)n(i)} \Big)$$

$$= g_i^{n(i)} \Big(\sum_{p:(n(i),p)\in \overline{A}} \overline{y}_{n(i)p} \underbrace{\sum_{j\in J_p} g_j^p}_{=1} - \sum_{p:(p,n(i))\in \overline{A}} \overline{y}_{pn(i)} \underbrace{\sum_{j\in J_p} g_j^p}_{=1} \Big)$$

$$= g_i^{n(i)} \Big(\underbrace{\sum_{p:(n(i),p)\in \overline{A}} \overline{y}_{n(i)p} - \sum_{p:(p,n(i))\in \overline{A}} \overline{y}_{pn(i)}}_{\overline{b}_{n(i)}} \Big)$$

$$= \frac{b_i}{\overline{b}_{n(i)}} \overline{b}_{n(i)}$$

$$= b_i$$

This holds for all sources and destinations (i.e. $\forall i \in S \cup D$). If i is an intermediate node we get in the third last row of the proof that $g_i^{n(i)} \overline{b}_{n(i)} = 0$ since $\overline{b}_{n(i)}$ is equal zero, hence the proof also holds $\forall i \in I$.

To ii.)

$$\overline{x}_{ij} = \underbrace{g_{ij}^{n(i)p(j)}}_{\geq 0} \overbrace{\overline{y}_{n(i)p(j)}}^{\geq 0} \leq g_{ij}^{n(i)p(j)} \underbrace{\overline{u}_{n(i)p(j)}}_{= \min\limits_{\substack{l\in J_{n(i)} \\ m\in J_{p(j)}}} \{ \frac{u_{lm}}{g_{lm}^{n(i)p(j)}} : g_{lm}^{n(i)p(j)} > 0 \}} \leq g_{ij}^{n(i)p(j)} \frac{u_{ij}}{g_{ij}^{n(i)p(j)}} = u_{ij}$$

b.)

$$\sum_{(i,j)\in A} c_{ij} \overline{x}_{ij}^* = \sum_{(i,j)\in A} c_{ij} g_{ij}^{n(i)p(j)} \overline{y}_{n(i)p(j)}^*$$

$$= \sum_{(n,p)\in \overline{A}} \overline{y}_{np}^* \underbrace{\sum_{\substack{i\in J_n \\ j\in J_p}} c_{ij} g_{ij}^{np}}_{=\overline{c}_{np}} = \sum_{(n,p)\in \overline{A}} \overline{c}_{np} \overline{y}_{np}^* = \overline{z}^*$$

q.e.d.

Corollary 5.2

$$z^* \leq \overline{z}^*$$

This means that solving the aggregated problem and disaggregating the corresponding solution yields an upper bound with value $\overline{z}^*$ for the original problem.

5.3.2 Bounds on the Loss of Accuracy

$\overline{z}^* - z^*$ expresses the loss of accuracy induced by solving the aggregated problem instead of the original one. As seen in the transportation problem, we can derive a bound on this quantity.

Proposition 5.4 [Zip80]

Given an original minimum cost network flow problem and a corresponding aggregated problem based on a partition $\overline{NP}$ of the node set and derived by using the weighted aggregation approach. Then the following inequality holds:

$$\overline{z}^* - z^* \le -\sum_{(n,p)\in\bar{A}} \bar{\alpha}_{np}(\bar{u}_{np} - \sum_{\substack{i\in J_n \\ j\in J_p}} u_{ij}) - \min_{x\in X} \sum_{(i,j)\in A} [c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)} - \bar{\alpha}_{n(i)p(j)})]\, x_{ij}$$

Proof:

$$z^* = \sum_{(i,j)\in A} c_{ij}x^*_{ij} = \sum_{(i,j)\in A} c_{ij}x^*_{ij} + \sum_{i\in N} \bar{\pi}_{n(i)} \underbrace{[b_i - (\sum_{j:(i,j)\in A} x^*_{ij} - \sum_{j:(j,i)\in A} x^*_{ji})]}_{=0}$$

$$\ge \sum_{(i,j)\in A} c_{ij}x^*_{ij} + \sum_{i\in N} \bar{\pi}_{n(i)}[b_i - (\sum_{j:(i,j)\in A} x^*_{ij} - \sum_{j:(j,i)\in A} x^*_{ji})] + \sum_{(n,p)\in\bar{A}} \underbrace{\bar{\alpha}_{np}}_{\ge 0} \underbrace{[\bar{u}_{np} - \bar{u}_{np}]}_{=0} + \underbrace{\sum_{\substack{i\in J_n \\ j\in J_p}} (x^*_{ij} - u_{ij})}_{\le 0}$$

$$= \underbrace{\sum_{n\in\bar{N}} \bar{\pi}_n \bar{b}_n - \sum_{(n,p)\in\bar{A}} \bar{\alpha}_{np}\bar{u}_{np}}_{=\overline{z}^*} + \sum_{(i,j)\in A} c_{ij}x^*_{ij} + \sum_{i\in N} \bar{\pi}_{n(i)}(-\sum_{j:(i,j)\in A} x^*_{ij} + \sum_{j:(j,i)\in A} x^*_{ji}) + \sum_{(n,p)\in\bar{A}} \bar{\alpha}_{np}[\bar{u}_{np} + \sum_{\substack{i\in J_n \\ j\in J_p}} (x^*_{ij} - u_{ij})]$$

$$= \overline{z}^* + \sum_{(i,j)\in A} c_{ij}x^*_{ij} - \sum_{(i,j)\in A} (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)})x^*_{ij} + \sum_{(i,j)\in A} \bar{\alpha}_{n(i)p(j)}x^*_{ij} + \sum_{(n,p)\in\bar{A}} \bar{\alpha}_{np}(\bar{u}_{np} - \sum_{\substack{i\in J_n \\ j\in J_p}} u_{ij})$$

$$= \overline{z}^* + \sum_{(i,j)\in A} [c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)} - \bar{\alpha}_{n(i)p(j)})]\, x^*_{ij} + \sum_{(n,p)\in\bar{A}} \bar{\alpha}_{np}(\bar{u}_{np} - \sum_{\substack{i\in J_n \\ j\in J_p}} u_{ij})$$

$$\ge \overline{z}^* + \min_{x\in X} \sum_{(i,j)\in A} [c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)} - \bar{\alpha}_{n(i)p(j)})]\, x_{ij} + \sum_{(n,p)\in\bar{A}} \bar{\alpha}_{np}(\bar{u}_{np} - \sum_{\substack{i\in J_n \\ j\in J_p}} u_{ij})$$

So finally we get:

$$\overline{z}^* - z^* \le -\sum_{(n,p)\in\bar{A}} \bar{\alpha}_{np}(\bar{u}_{np} - \sum_{\substack{i\in J_n \\ j\in J_p}} u_{ij}) - \min_{x\in X} \sum_{(i,j)\in A} [c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)} - \bar{\alpha}_{n(i)p(j)})]\, x_{ij}$$

q.e.d

In order to calculate the bound derived above, it would be necessary to solve a problem of the same size as the original problem. If we apply the same concepts already used in the last section (i.e. dropping the supply as well as all capacity constraints) we finally get the following bound.

A posteriori bound on the loss of accuracy for the weighted aggregation approach [Zip80]

$$\bar{z}^* - z^* \le -CAP - \sum_{d \in D} |b_d| SP_d \qquad (5.2)$$

With

$$CAP = \sum_{(n,p) \in \bar{A}} \bar{\alpha}_{np} (\bar{u}_{np} - \sum_{\substack{i \in J_n \\ j \in J_p}} u_{ij})$$

$$LE_{ij} = c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)} - \bar{\alpha}_{n(i)p(j)})$$

SP_d = the shortest path from any source to destination $d \in D$, with LE_{ij} as the length between node *i* and *j*, $(i, j) \in A$

CAP can be calculated easily because $\bar{u}_{np}$ and u_{ij} are known in advance and $\bar{\alpha}_{np}$ can be derived by solving the aggregated problem. It requires more effort to calculate the shortest path SP_d from any source to destination $d \in D$. In contrast to the bound (5.1) presented in the last section two possibilities exist to reduce this effort: the first possibility is a kind of relaxation whereas the second one utilizes the setting of a special case.

So let us have a closer look on the first possibility. For this, we define the length of an arc $(n, p) \in \bar{A}$ in the aggregated network as

$$\overline{LE}_{np} = -(\bar{\pi}_n - \bar{\pi}_p - \bar{\alpha}_{np}) + \min\{c_{ij} : i \in J_n, j \in J_p\}$$

Further, let us denote with $\overline{SP}_{\bar{d}}$ the shortest path from any source to $\bar{d} \in \bar{D}$ in the aggregated network using the arc length defined above.

Obviously, the following inequality holds

$$\bar{z}^* - z^* \le -CAP - \sum_{d \in D} \bar{b}_{\bar{d}} \overline{SP}_{\bar{d}} \qquad (5.3)$$

As we will see later, while presenting the second possibility for reducing computational effort, the bound derived above leads to the same result as the original bound, if all destinations left unaggregated. In general, the derived bound is somewhat looser than the original one. But the computation may avoid certain setup costs and computational effort because the aggregated network is already defined and smaller than the original one. Of course it is possible to derive a similar expression for bound (5.1) derived in the last section. But then we would lose much more accuracy than applying it in Zipkin's setting. The reason for this is that the bound derived above utilizes the assumptions (AZ 2) and (AZ 3), which are not satisfied for general aggregations.

The second possibility utilizes the setting of a special case. If we assume that all destinations left unaggregated and define the length of an arc $(n, p) \in \overline{A}$ as $\overline{LE}_{np}$, then the corresponding shortest path $\overline{SP}_{\bar{d}}$ from any source to destination $\bar{d} \in \overline{D}$ is equal SP_d. This holds, since we know from assumption (AZ 2) that there are no arcs between nodes $i, j \in N$ in a subset J_n and we get from assumption (AZ 3) that whenever an arc $(n, p) \in \overline{A}$ between node *n* and node *p* exists, then $(i, j) \in A$ holds $\forall i \in J_n, j \in J_n$. Therefore, if we assume that all destinations left unaggregated as well as that we have no cycles with negative weight in the aggregated network, bound (5.3) leads to the same result as bound (5.2).

As we have seen, an a posterior bound can be derived based on the same concepts also applied to the transportation problem. The question remains, if we can also do so in order to derive an a priori bound ? Therefore let us again have a look at the following expression:

$$-\sum_{(n,p)\in\bar{A}} \bar{\alpha}_{np}(\bar{u}_{np} - \sum_{\substack{i\in J_n \\ j\in J_p}} u_{ij}) - \min_{x\in X} \sum_{(i,j)\in A} [c_{ij} - (\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)} - \bar{\alpha}_{n(i)p(j)})]x_{ij}$$

To obtain an a priori bound for the second term of this expression, we can substitute the expression in brackets with $(c_{ij} - \bar{c}_{n(i)p(j)})$. This holds, since $(\bar{\pi}, \bar{\alpha})$ is a feasible pair for the dual problem, hence $\bar{\pi}_{n(i)} - \bar{\pi}_{p(j)} - \bar{\alpha}_{n(i)p(j)} \leq \bar{c}_{np}$. By dropping the flow conservation constraints for all supply nodes as well as the capacity constraints, we would finally get an a priori bound for the second term. The derivation of a bound for the whole term would also require upper bounds on the $\bar{\alpha}_{np}$. We tried to derive such bounds but, as Zipkin, we failed. So the a posterior bound derived already is the only bound which could be presented for the aggregation of minimum cost network flow problems using the weighted aggregation approach.

Remark:

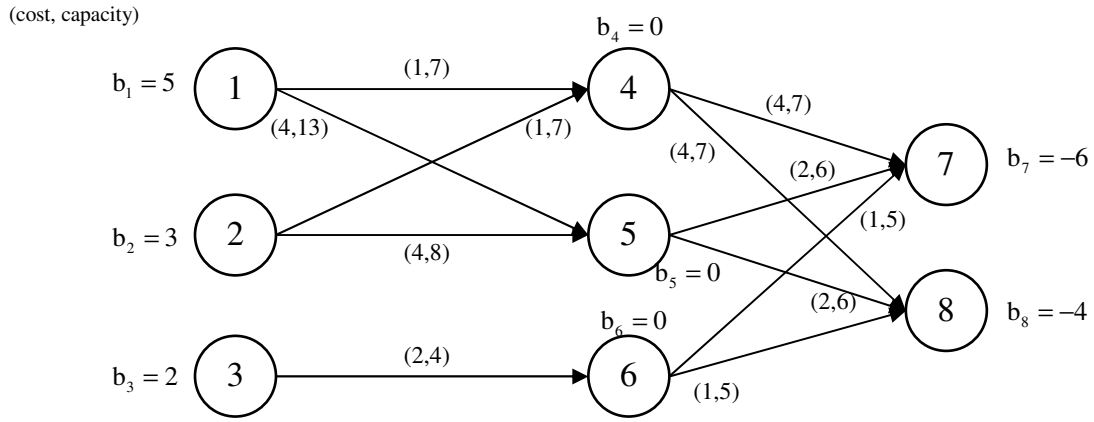
1.) The observations made concerning the a priori bound are also valid for the aggregation by dominance approach.

2.) By dropping the demand constraints instead of the supply constraints a second a posteriori bound can be derived. Instead of computing the shortest path from any source to each destination $d \in D$, we would have to calculate the shortest path from each source $s \in S$ to any destination to in order to derive the bound.

Before we close this section with a discussion of the advantages and drawbacks of Zipkin's approach, we will have a look at a concluding example. In it we include an example for the calculation of the bound derived in the last section as well.

Example 5.8

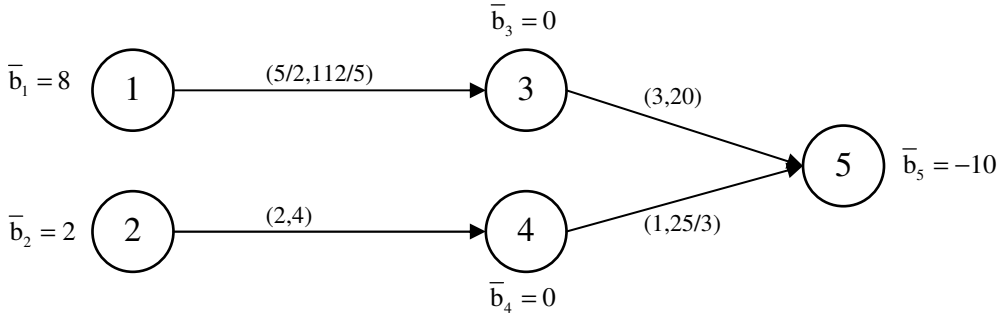
UAMCNFP



The UAMCNFP defined above has an optimal objective value of $z^* = 46$

The following aggregated problem is based on a partition $\overline{NP}$ of the node set, which consists of $J_1 = \{1,2\};\ J_2 = \{3\};\ J_3 = \{4,5\};\ J_4 = \{6\};\ J_5 = \{7,8\}$ and satisfies (AZ1-3):

$(\overline{c}_{conv}, \overline{u}_{min})$

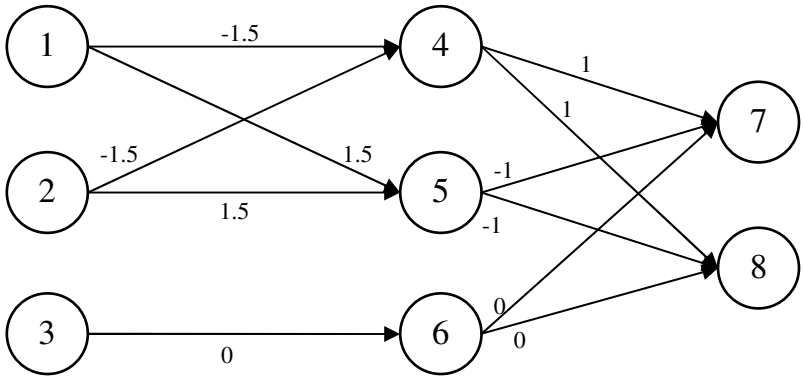


$\overline{z}^* = 50$ is the optimal objective value of the aggregated problem.

To calculate the a posterior bound, we need the optimal dual pair $(\overline{\pi}, \overline{\alpha})$ of the aggregated problem first:

$\overline{\pi}_1 = 5.5;\ \overline{\pi}_2 = 3;\ \overline{\pi}_3 = 3;\ \overline{\pi}_4 = 1;\ \overline{\pi}_5 = 0; \qquad \overline{\alpha}_{13} = \overline{\alpha}_{24} = \overline{\alpha}_{35} = \overline{\alpha}_{45} = 0$

The following network can be used to calculate the shortest path from any source to each destination:



Finally we get:

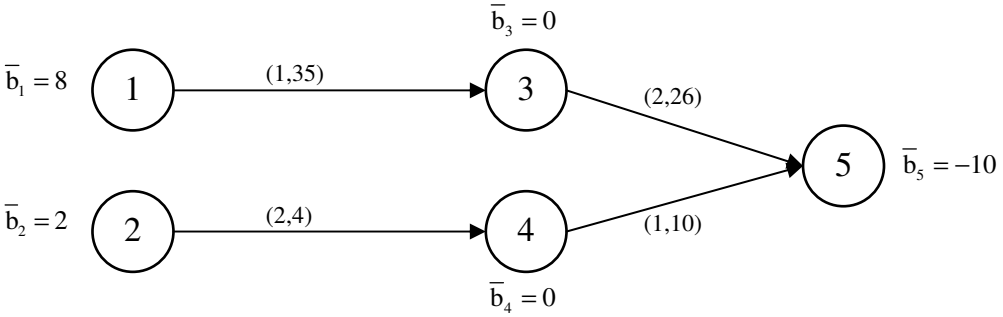
$$\overline{z}^* - z^* = 50 - 46 = 4 \le -CAP - \sum_{d \in D} |b_d| \, SP_d = 0 - [6*(-0.5) + 4*(-0.5)] = 5$$

For the second a posteriori bound we get:

$$\overline{z}^* - z^* = 50 - 46 = 4 \le -CAP - \sum_{s \in S} b_s SP_s = 0 - [5*(-0.5) + 3*(-0.5) + 2*0] = 4$$

So let us calculate the bound derived in Section 5.2, in which we used $\overline{c}_{min}$ and $\overline{u}_{\Sigma}$ as respecification maps. Therefore the aggregated problem, based on a partition $\overline{NP}$ defined on the previous page, has the following parameters:

$(\overline{c}_{\min}, \overline{u}_{\Sigma})$

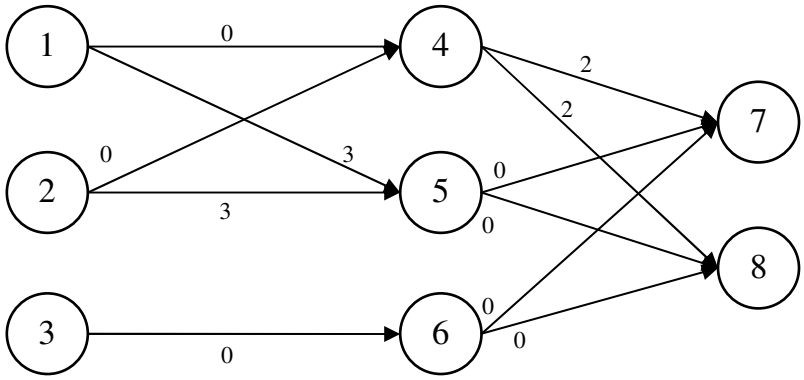


The aggregated problem using $\overline{c}_{min}$ and $\overline{u}_{\Sigma}$ as respecification maps has an optimal objective value of $\overline{z}^* = 30$.

In order to calculate the a posterior bound, we need the optimal dual pair $(\overline{\pi}, \overline{\alpha})$ first.

$$\overline{\pi}_1 = 3;\ \overline{\pi}_2 = 3;\ \overline{\pi}_3 = 2;\ \overline{\pi}_4 = 1;\ \overline{\pi}_5 = 0; \qquad \overline{\alpha}_{13} = \overline{\alpha}_{14} = \overline{\alpha}_{23} = \overline{\alpha}_{24} = \overline{\alpha}_{35} = \overline{\alpha}_{45} = 0$$

The following network can be used to calculate the longest path from any source to each destination:



Finally we get:

$$z^* - \overline{z}^* = 46 - 30 = 16 \leq \sum_{d \in D} |b_d| LP_d + \sum_{(n,p) \in \overline{A}} \overline{\alpha}_{np} \overline{u}_{np} = (6*3 + 4*3) + 0 = 30$$

For the second a posteriori bound we get:

$$z^* - \overline{z}^* = 46 - 30 = 16 \leq \sum_{s \in S} b_s LP_s + \sum_{(n,p) \in \overline{A}} \overline{\alpha}_{np} \overline{u}_{np} = (5*3 + 3*3 + 2*0) + 0 = 24$$

5.3.3 Advantages and Drawbacks of the Weighted Aggregation Approach

So let us conclude this section by giving some advantages and drawbacks of Zipkin's approach.

Advantages:

- The fixed-weight disaggregation provides us very quickly with a feasible solution for the original problem
- The weighted aggregation yields an upper bound on the optimal objective value of the original problem
- Based on the derived a posteriori bound we can decide, if the result of solving the AMCNFP is good enough for our application or if it is necessary to recalculate

Drawbacks:

- We need more effort to set up the costs and capacities of the aggregated problem as it was the case when we applied $\overline{c}_{min}$ and $\overline{u}_{\Sigma}$ as respecification maps

- In most cases the fixed-weight disaggregation will lead to non-integral solutions for the UAMCNFP. However, for most real world applications an integral solution is required.

- The definition of the capacity for the aggregated problem is very restrictive and results often in an infeasible AMCNFP. It is possible to have less restrictive respecification maps for the capacity, such as $\overline{u}_{\Sigma}$, but then the fixed-weight solution might be infeasible. In such a case the disaggregation has to be handled in a more analytical way resulting in more effort and destroying the advantage of the fixed-weight disaggregation as a very efficient and quick disaggregation map. See for example [Zip77] for detailed information.

- To apply the weighted aggregation approach several assumptions have to be satisfied. Some of them are very restrictive for the application to real world problem instances

The usefulness of applying $\overline{c}_{conv}$ and $\overline{u}_{min}$ for generating the costs and capacities of the aggregated problem mainly depends on the application. We saw this in the last section in which we used $\overline{c}_{min}$ and $\overline{u}_{\Sigma}$ as respecification maps, too. If it is not necessary to calculate an optimal solution for the original problem and only an upper bound on the objective value is required as well as the stated assumptions can be satisfied, the presented approach is very valuable. In the case that a distribution of flow for the original problem is required, things become even more complicated, since most real world applications require an integral solution.

5.4 Measures on Aggregation and the Grouping of Nodes

In the following last section of Chapter 5 we will take up and extend ideas found in literature regarding the degree of aggregation and how the grouping of nodes can be reasonable implemented.
The *degree of aggregation* should provide us with a value for the degree of discrepancy, qualitative and quantitative, between the aggregated and the unaggregated problem.
If we compare qualitative and quantitative measures, we could observe that a quantitative measure provides only information about the structural differences of both networks, whereas a qualitative measure for the degree of aggregation provides also information about the nodes\arcs which are combined together, concerning their similarity (e.g. successors, predecessors, costs, etc.).
A possible quantitative degree of aggregation is the size of the aggregated problem relative to the size of the original one. Therefore we use several combinations of this relation for the following measures.

$$\text{QUANTDA}_{\text{I}} = \frac{|\bar{\text{N}}| + |\bar{\text{A}}|}{|\text{N}| + |\text{A}|} \qquad \in [0,1]$$

$$\text{QUANTDA}_{\text{II}} = \frac{|\bar{\text{N}}|}{|\text{N}|} + \frac{|\bar{\text{A}}|}{|\text{A}|} \qquad \in [0,2]$$

$$\text{QUANTDA}_{\text{III}} = \frac{|\text{N}| - |\bar{\text{N}}|}{|\text{N}| - 1} + \frac{|\text{A}| - |\bar{\text{A}}|}{|\text{A}|} \qquad \in [0,2]$$

For $|\text{N}| > 1, |\bar{\text{N}}| \geq 1$ and $|\text{A}| \geq 1$;

For measure I and II, we observe that the more nodes and arcs are aggregated the closer the corresponding value will be to zero, whereas for measure III the opposite is true. Even though the presented measures are very simple, they provide a first step for the characterization of aggregated networks. But as already mentioned the measures defined above do not contain any qualitative information.
There are lots of possibilities grouping the nodes together. Most of them are out of question, since they lead to an unrealistic aggregated problem (e.g. no one would group two warehouses or rooms together, which are far away from each other). In order to decide which of the reasonable partitions should be preferred, qualitative measures can provide information in particular. The following measure compares the similarity regarding the different predecessors and successors of the grouped nodes and can therefore be classified as a qualitative measure.

$$QUALDA_I = \sum_{J_n \in \overline{NP}} \frac{\left|\bigcap_{i \in J_n} pred(i)\right| + \left|\bigcap_{i \in J_n} succ(i)\right|}{\left|\bigcup_{i \in J_n} pred(i)\right| + \left|\bigcup_{i \in J_n} succ(i)\right|} \quad \in [0, m_{\overline{NP}}]$$

A value close to $m_{\overline{NP}}$ corresponds to an aggregation, in which the nodes grouped in each subset have entirely the same predecessors and successors. If we assume that assumption (AZ 3) of Zipkin holds, we would get a value of $m_{\overline{NP}}$ for $QUALDA_I$. Of course this observation also holds if no arcs at all are aggregated.

While the measures QUANTDA I-III only provide us with some quantitative information concerning the aggregation, $QUALDA_I$ provides us with further information about the inherent structure of aggregation. Measures like QUANTDA I-III can be used in aggregation-algorithms as a stop or termination criterion. This means that the algorithm groups nodes together until some degree of aggregation is satisfied. $QUALDA_I$ in contrast can be used to determine how the nodes should be grouped together. Before we go on with an algorithm that can be used for grouping nodes together corresponding to the defined measures, let us have a look on an example in which we calculated the different measures for the aggregation of Example 5.4.

Example 5.9

$$QUANTDA_I = \frac{11}{17}$$

$$QUANTDA_{II} = \frac{31}{24} \qquad QUALDA_I = \frac{0}{3} + \frac{2}{2} + \frac{1}{6} + \frac{3}{3} + \frac{0}{3} = 2\frac{1}{6}$$

$$QUANTDA_{III} = \frac{16}{21}$$

Zipkin [Zip80] derived an algorithm for the partitioning of nodes in the case of the transportation problem. His method comprises systematic refinement of initial partitions. He used the derived bounds, seen in Chapter 4, as a decision criterion for the refinement of the partitions. His approach is very restrictive because the only criterion for grouping depends on the particular costs of the arcs, but all told, it seems to be reasonable for the transportation problem. The following algorithm takes besides the parameters also the particular structure of the original network into account and is based on the basic concepts of Lee [Lee75].

Algorithm for Deriving a Partition $\overline{NP}$ of the Node Set

INPUT: UAMCNFP, $\varepsilon \in [0,1]$

OUTPUT: Partition $\overline{NP}$ of the node set N

STEP 1: Set

$$k = 1$$
$$\overline{NP} = \varnothing$$

STEP 2:

$$J_k = \{i \in N : pred(i) = \varnothing\}$$
$$\overline{NP} = \overline{NP} \cup J_k$$
$$N = N \setminus J_k$$
$$A = A \setminus \{i, j\} \text{ where } i \in J_k, j \in succ(i)$$
$$k = k + 1$$

If $N = \varnothing$ → STOP

Else → Repeat STEP 2

STEP 3:

For each $J_k \in \overline{NP}$

If $$\left(\frac{\left| \bigcap_{i \in J_k} pred(i) \right| + \left| \bigcap_{i \in J_k} succ(i) \right|}{\left| \bigcup_{i \in J_k} pred(i) \right| + \left| \bigcup_{i \in J_k} succ(i) \right|} \right)^{*} \geq \varepsilon \quad \text{then}$$

Split J_k up into new subsets J_{k_l} until * holds for all such subsets J_{k_l}

$$\overline{NP} = \overline{NP} \setminus J_k$$
$$\overline{NP} = \overline{NP} \cup \{J_{k_1}, ..., J_{k_n}\}$$
$$\text{with } J_k = J_{k_1} \cup ... \cup J_{k_n} \text{ and } J_{k_q} \cap J_{k_r} = \varnothing \; \forall q \neq r$$

Else leave J_k as it is.

Remark: In STEP 3 alternative measures for refining J_k can also be used.

The presented algorithm combines the quantitative and the qualitative point of view concerning aggregation as well. The algorithm can also be used for the partitioning of nodes in the case of the transportation problem.

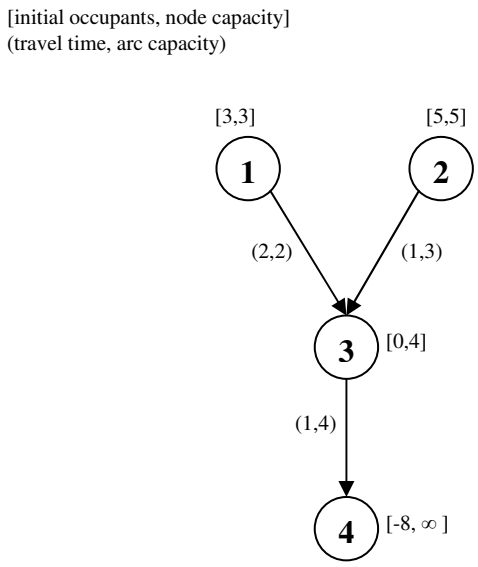
One of the main advantages of it lies in the preserving of the general chain flows in the aggregated network where also some of the fine details in local regions are suppressed (see STEP 2). We stress again the fact that the aggregation of an original problems, depends to a high degree on the underlying application. The theory given so far can be seen as a starting point of aggregation. However, it has to be adjusted according to the particular application, as we will see in the following chapter about the application to a real word problem.

Chapter 6

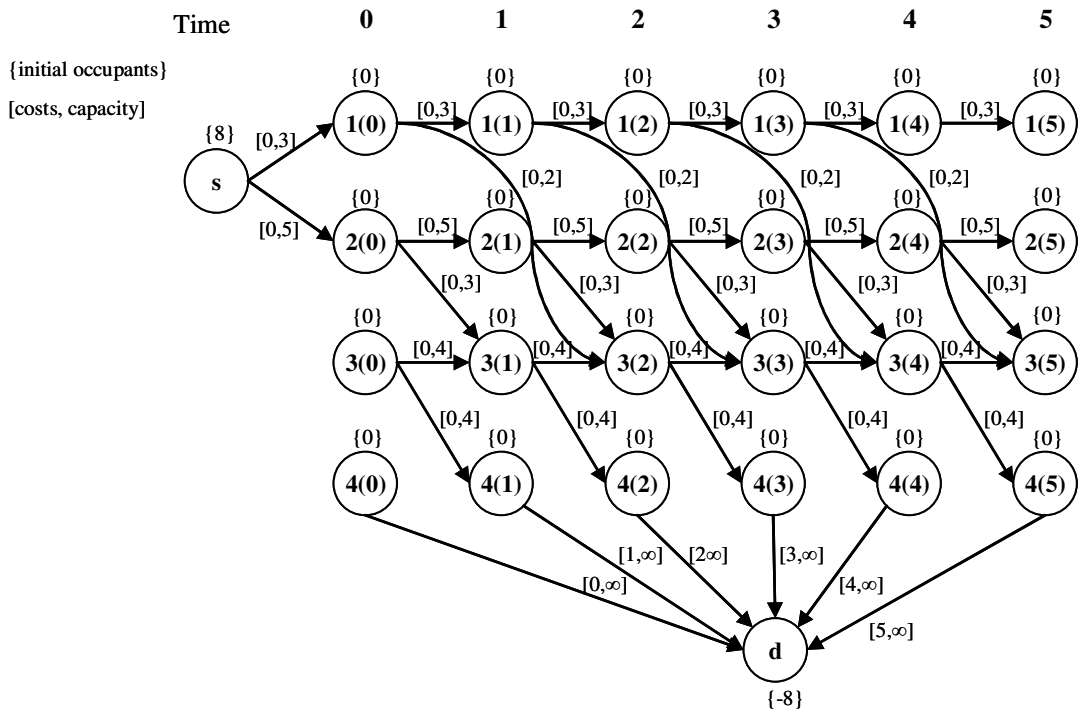
Aggregation of the Evacuation Problem

In this chapter, we are going to discuss aggregation of evacuation problems, defined in Chapter 2. The evacuation problem was based on a dynamic network $G_{DYN} = (N, A, T)$, which consists of a static network $G_{STA} = (N, A)$ and a time horizon T. It is possible to represent it as an equivalent static minimum cost network flow problem in the time expanded network. Unfortunately, the size of the time expanded network depends on the time horizon T and grows very fast. In order to solve evacuation problems of larger buildings, we decided to apply aggregation. Because of the fact that we can represent the evacuation problem as a static MCNFP, we will examine in the following if it makes sense to apply the aggregation directly to the time expanded network or if it is more advisable to apply the aggregation to $G_{DYN} = (N, A, T)$ before deriving the time expanded network. We will also have a look on how the concepts discussed in the last chapter can be applied for the dynamic case, too.
In order to recall the connection between the dynamic- and time expanded network, the following example shows an original (unaggregated) evacuation problem and the corresponding representation as a static MCNFP in the time expanded network. For the original problem, we assume a basic time unit $\pi = 1$ (i.e. the length of one time period is equal to one second).

Example 6.1



The dynamic network $G_{DYN} = (N, A, T)$ that is defined above corresponds to an evacuation problem. If we assume a time horizon $T = 5$, we get the following time expanded representation for the evacuation problem.



The time expanded network can be divided into two structural dimensions. The first one is the *horizontal dimension*. In the horizontal dimension, each node $i \in N$ of the dynamic network has a time copy for each time unit. Between two successive node copies (e.g. $i(t)$ and $i(t+1)$) we have at most one arc, representing the holdover flow. The horizontal dimension gives information on how much flow (people) stays at a node from time unit t to $t+1$.
The second dimension is the *vertical dimension*. The vertical dimension represents the status of all "original" nodes at a particular time unit. Status means, how much flow (people) arrives at this node at time *t* and how much flow leaves this node at time *t*.
Our discussion about the aggregation of the evacuation problem will also be divided into these two dimensions.
The following figure shows the two dimensions of the time expanded network of Example 6.1.

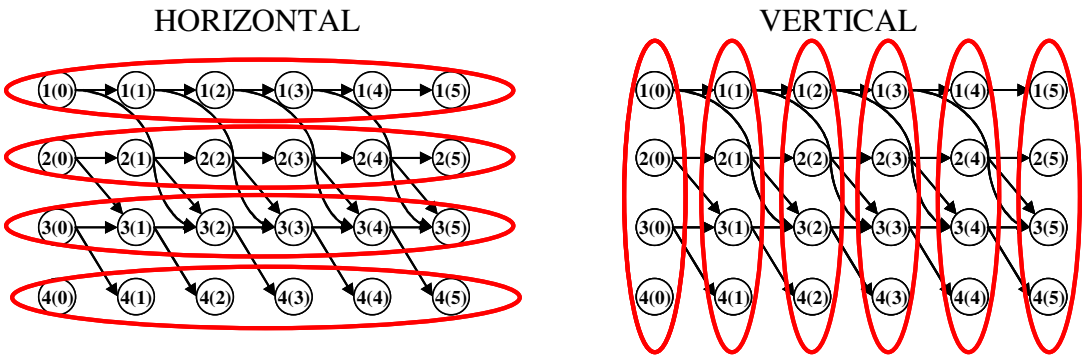


Figure 17: Horizontal and vertical dimension of the time expanded network of Example 6.1

In the following section we describe the horizontal aggregation of evacuation problems. The second possibility of aggregating evacuation problems, the vertical aggregation, is described in Section 6.2. In Section 6.3 we discuss the application of the respecification maps that are already presented in Chapter 4/5 and present a different approach for calculating the required

parameters, adjusted to the evacuation problem. As already observed for the transportation problem and the minimum cost network flow problem the aggregation of the evacuation problem will also lead to a loss of accuracy, which is discussed in Section 6.4. We provide a new theoretical result on this loss for a special case of aggregation. The chapter is concluded with some empirical tests about the impact of aggregation applied to the evacuation problem. Most of the results we got in this chapter are also valid for dynamic network flow problems in general.

6.1 Horizontal Aggregation

The first kind of aggregation that we are going to discuss is the aggregation in the horizontal dimension. Horizontal aggregation means that we aggregate only nodes which are time copies of the same original node. In a more formal way this can be expressed as follows:

Horizontal Aggregation

$$i(t) \text{ and } i(t') \text{ are aggregated} \Leftrightarrow i(t), i(t') \in \{i(0),...,i(t),...,i(T)\}, i \in N$$

The nodes aggregated together are time copies of an original node $i \in N$; therefore we have to take the following assumptions into account.

(AS1) $i(t)$ and $i(t')$ are grouped together

$$\Rightarrow \quad t' = t+1$$

(AS2) $i(t)$ and $i(t+1)$ are grouped together

$$\Rightarrow \quad i(k) \text{ and } i(k+1) \text{ for } k = 0...t-1 \text{ are grouped together}$$
$$i(k) \text{ and } i(k+1) \text{ for } k = t+2,...,T \text{ are grouped together}$$

(AS3) $i(t)$ and $i(t+1)$ are grouped together satisfying AS1 and AS2

$$\Rightarrow \quad j(t) \text{ and } j(t+1) \text{ are grouped together } \forall j \in N, t = 0,...,T$$

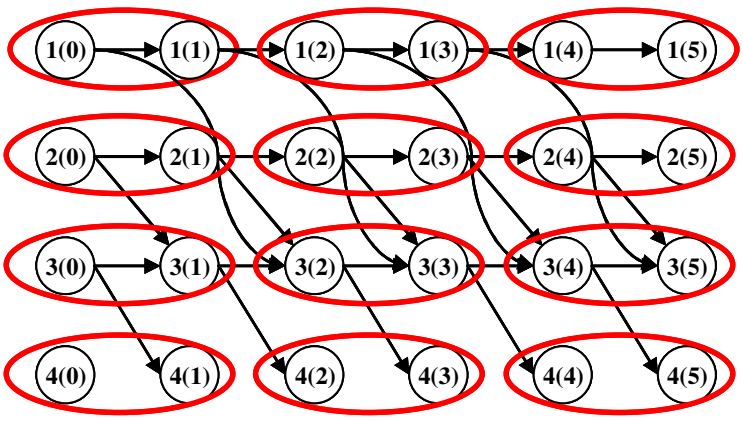


Figure 18: Horizontal aggregation satisfying (AS 1-3)

The figure above shows a horizontal aggregation satisfying the stated assumptions.

At first glance, the assumptions made so far seem to be very restrictive. They might be compared with the assumptions of Zipkin for which we annotated a strong restriction of reality. However, unlike Zipkin's assumptions they are absolutely necessary. We must be aware of the fact that the time expanded network is not a matter of a general static network, but an equivalent representation of a dynamic network. If we omitted one of the assumptions, we would lose the equivalence to the dynamic network, because all the nodes and arcs of the time expanded network represent an original node/arc for a particular time *t*. Suppose we merge node $3(0)$ and $3(5)$ of the time expanded network presented in the last figure and leave the nodes $3(t), t = 2,3,4$ unaggregated i.e. violating (AS1). If we make such an aggregation we would suggest some leap in time and would have no reasonable interpretation in terms of the dynamic network.
The observations made so far lead us to the perception that it would make more sense to apply the horizontal aggregation directly to the evacuation problem before constructing the time expanded network. Horizontal aggregation in the time expanded network is the same as introducing time units $\overline{t}$ corresponding to a basic time unit π unequal one second in the evacuation problem. This means that we increase the length of one time period. This is a well known approach for reducing the size of dynamic networks. Because of the fact that, we only change the parameter *T* of the dynamic network $G_{DYN} = (N, A, T)$ in a horizontal aggregation, the following definition for the corresponding aggregated evacuation problem holds.

Horizontal Aggregated Evacuation Problem

(HAEVAC)

$$\min \overline{T}$$

s.t.

$$x_{ii}(\overline{t}-1) - x_{ii}(\overline{t}) = \sum_{j:(i,j)\in A} x_{ij}(\overline{t}) - \sum_{j:(j,i)\in A} x_{ji}(\overline{t}-\overline{\lambda}_{ji})$$
$$\overline{t} = 0,1,...,\overline{T}; \forall i \in N \setminus \{s,d\}$$

$$\sum_{\overline{t}=0}^{\overline{T}} \sum_{i\in D} x_{id}(\overline{t}) = \sum_{i\in S} EV_i$$

$$x_{si}(0) = EV_i, \qquad \forall i \in S$$

$$x_{ii}(\overline{t}) = 0, \qquad \forall i \in D; \overline{t} = 0,1,...\overline{T}$$

$$0 \le x_{ii}(\overline{t}) \le h_i, \qquad \overline{t} = 0,1,...,\overline{T}; i \in N \setminus D$$

$$0 \le x_{ij}(\overline{t}) \le \overline{u}_{ij}, \qquad \overline{t} = 0,1,...,\overline{T} - \overline{\lambda}_{ij}; \forall (i,j) \in A$$

Remark: The horizontal aggregation reduces only the time horizon, by combining seconds to time periods; this means that the node and arc set is the same as in the original problem.

The following algorithm shows the proceeding of horizontal aggregation applied directly to the dynamic network. The algorithm also includes a step in which the respecification maps for the horizontal aggregation are defined.

Algorithm for the Horizontal Aggregation

INPUT: Original evacuation problem, with travel times measured in seconds, capacity denoted in persons per second and a new basic time unit π

OUTPUT: Corresponding aggregated evacuation problem, based on a horizontal aggregation with a basic time unit π

STEP 1: $\overline{T} = \left\lceil \frac{T}{\pi} \right\rceil$

STEP 2:

For each arc $(i, j) \in A$ Do

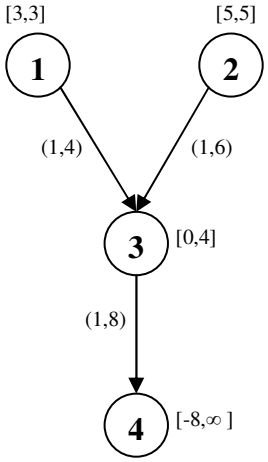
$$\overline{\lambda}_{ij} = \left\lceil \frac{\lambda_{ij}}{\pi} \right\rceil$$

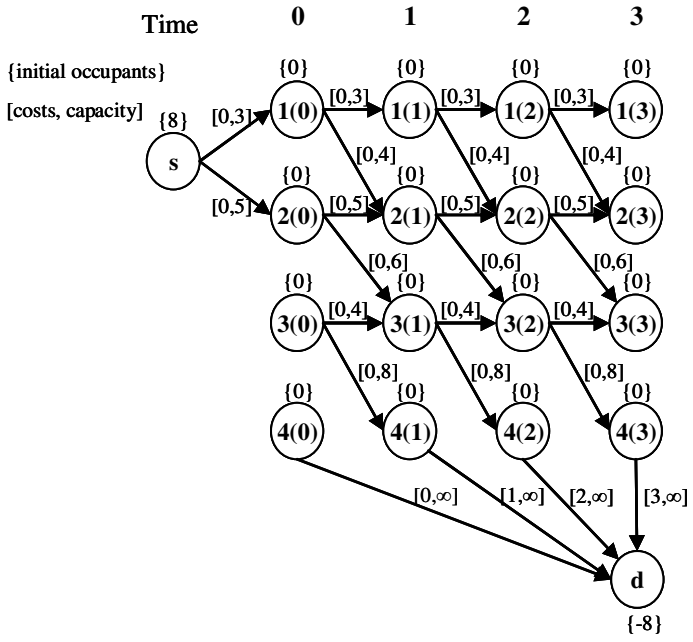
$$\overline{u}_{ij} = u_{ij}\pi$$

Remark: The capacity of an arc does not bound the total flow on that arc at a given time *t,* but defines the maximum number of flow that can enter the arc at each time *t*. Therefore we have to multiply the original capacity by the basic time unit π, in order to derive the capacity for the aggregated problem in STEP 2.

In the following example we have applied the algorithm to the evacuation problem of Example 6.1, where we take a basic time unit $\pi = 2$.

Example 6.2





As mentioned before, a change of the basic time unit is a well known approach for reducing the size of a dynamic network in general. In order to get a solution for the original problem in seconds, the optimal evacuation time of the aggregated problem is multiplied by the basic time unit π. However, as it is often the case in aggregation, this kind of aggregation introduces a loss of accuracy as well. We will discuss the evaluation of the error caused by solving the aggregated problem instead of the original problem in a separate section, since it is an important topic regarding aggregation. So let us continue with the second kind of aggregation for the evacuation problem.

6.2 Vertical Aggregation

As seen in the last section, it was more reasonable to apply the horizontal aggregation directly to the dynamic network instead of the time expanded network. Nevertheless, we also try to apply the vertical aggregation to the time expanded network first. The vertical aggregation allows us to aggregate node copies which belong to different original nodes $i, j \in N$.

Vertical Aggregation

$$i(t) \text{ and } j(t') \text{ are aggregated} \Leftrightarrow i, j \in N : i \neq j$$

At first view, the vertical aggregation allows us various possibilities for grouping nodes together. Such an aggregation can be compared with the aggregation already seen for the transportation or the minimum cost network flow problem. Again, we have to remember that the time expanded network is an equivalent representation of a dynamic network. Therefore we have to take the following assumptions into account:

(AS4) Two nodes $i(t)$ and $j(t')$ are grouped together

$$\Rightarrow \quad t = t'$$

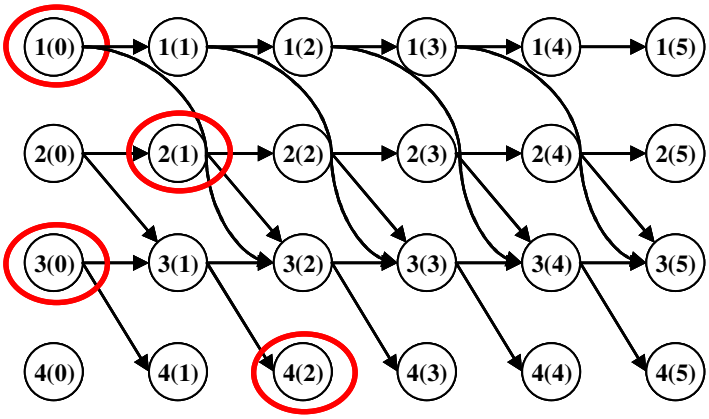


Figure 19: Vertical aggregation violating assumption (AS4)

A vertical aggregation applied to the nodes expressed by the circles in the figure above would violate assumption (AS4), because they present different time units.

(AS5) Two nodes *i(t)* and *j(t)* are grouped together

$\Rightarrow$ $i(t')$ and $j(t')$ are grouped together $t' \in [0,t) \cup (t,T]$

The following figure shows a vertical aggregation satisfying assumptions (AS4) and (AS5)

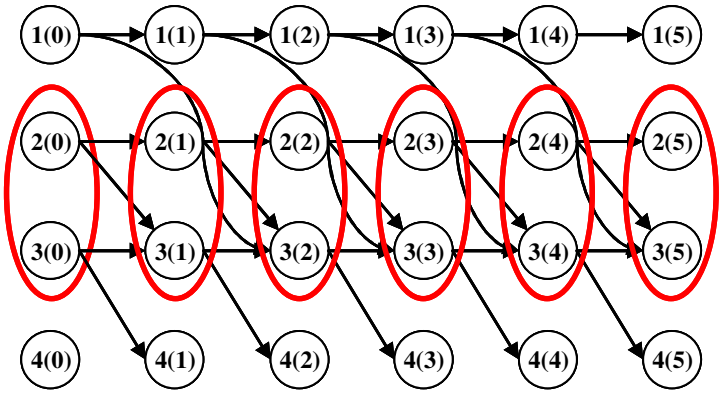


Figure 20: A vertical aggregation satisfying (AS 4-5)

As it is the case in the horizontal aggregation the assumptions are absolutely necessary. If we violate, for example, assumption (AS 4) and merge two nodes *i(t)* and $j(t')$, with $t \neq t'$, the first problem we are faced with is the assignment of the new aggregated node to a time unit (e.g. t or t'). We would also lose the equivalence to the dynamic network, because the nodes represent the status of an original node $i \in N$ for a particular time *t*. Again, it is advisable to apply the vertical aggregation directly to the dynamic network. In the case of the vertical aggregation, the aggregation can be applied to G_{STA}. We get a formal definition of the vertical aggregated evacuation problem, when we recall the definition of $\overline{NP}$, the partition of the node set *N*, first.

Recall (Definition 5.1)

Let $\overline{\mathrm{NP}} = \{\mathrm{J_k} : \mathrm{J_k} \subseteq \mathrm{N}\}$ be a partition of the node set N satisfying

i.) $\bigcup_{\mathrm{J_k} \in \overline{\mathrm{NP}}} \mathrm{J_k} = \mathrm{N}$

ii.) $\mathrm{J_k} \cap \mathrm{J_l} = \varnothing \qquad \forall\, \mathrm{J_k}, \mathrm{J_l} \in \overline{\mathrm{NP}},\ \mathrm{k} \neq \mathrm{l}$

$\overline{NP}$ has the same interpretation already discussed in Chapter 5.

If a partition $\overline{NP}$ of the node set *N* of an evacuation problem is given, the vertical aggregated evacuation problem can be described as follows:

Vertical Aggregated Evacuation Problem

(VAEVAC)

$$\min T$$

s.t.

$$\overline{y}_{nn}(t-1) - \overline{y}_{nn}(t) = \sum_{p:(n,p)\in\overline{A}} \overline{y}_{np}(t) - \sum_{p:(p,n)\in\overline{A}} \overline{y}_{pn}(t-\tilde{\lambda}_{pn})$$

$$t = 0,1,...,T; \forall n \in \overline{N} \setminus \{s,d\}$$

$$\sum_{t=0}^{T} \sum_{n\in\overline{D}} \overline{y}_{nd}(t) = \sum_{n\in S} \widetilde{EV}_n$$

$$\overline{y}_{sn}(0) = \widetilde{EV}_n, \qquad \forall n \in \overline{S}$$

$$\overline{y}_{nn}(t) = 0, \qquad \forall n \in \overline{D}; t = 0,1,...T$$

$$0 \le \overline{y}_{nn}(t) \le \tilde{h}_n, \qquad t = 0,1,...,T; i \in \overline{N} \setminus \overline{D}$$

$$0 \le \overline{y}_{np}(t) \le \tilde{u}_{np}, \qquad t = 0,1,...,T - \tilde{\lambda}_{np}; \forall (n,p) \in \overline{A}$$

Let us denote with $\overline{T}^*$ an optimal solution for the aggregated problem. For the flow $\overline{y}$ in the aggregated problem the Definition 2.1 of the dynamic flow already seen in Chapter 2 holds. For the meaning of the other parameters we also refer to Chapter 2.

The algorithm for aggregating minimum cost network flow problems saw in Chapter 5 can also be applied for the evacuation problem. The vertical aggregation in the time expanded network which is marked in the last figure can be derived by applying this algorithm to G_{STA} and using the following partition of the node set.

$$\overline{NP} = \{J_1, J_2, J_3\} \text{ with } J_1 = \{1\}, J_2 = \{2,3\}, J_3\{4\}$$

Of course, the respecification maps defined in Chapter 5 can also be used for dynamic network flow problems. However, in the following section we will discuss a different approach for deriving the parameters of the aggregated problem. This approach takes the characteristics of our problem instance into account.
Finally, we should point out that the aggregation applied to the dynamic network, before deriving the time expanded network, causes the same reductions of the network size as applying the aggregation directly to the time expanded network. We will assume in the following that the assumptions (AS 1-5) for the horizontal- and vertical aggregation, respectively, are satisfied.

6.3 Aggregation applied to the Real World Problem Instance

We have seen that we can distinguish between two dimensions concerning the aggregation of the evacuation problem; namely the *horizontal* and the *vertical dimension*. It makes no sense to apply the horizontal aggregation or the vertical aggregation directly to the time expanded network. It is advisable to apply it to the dynamic network first, before deriving the time expanded network. The application of the aggregation to the dynamic network results in the same reduction of the network size for the time expanded network and satisfies the stated assumptions.
The assumptions and observations made so far are valid for general evacuation problems. As we have seen in the chapters on aggregation theory before, the aggregation mainly depends on the particular problem instance. Therefore, we will discuss in the following the problem characteristics regarding the aggregation of the dynamic network representation of the Office Complex and the Casino of the EVZ.

6.3.1 Necessary Assumptions for the Aggregation of the given Problem Instance

We have seen in Chapter 3, that the modeling of evacuation objects means, next to other things, that nodes represent different functional segments of a building. So let us recall which segments (locations) of the Office Complex and the Casino are represented by nodes.

- Rooms (offices, rest rooms, stockrooms)
- Hallway segments
- Virtual rooms given by the groups of tables in the Casino
- Virtual hallway segments given by the positioning of tables and other obstructions (e.g. flowerbeds) in the Casino
- Safety areas

We assume that occupants are located in the rooms (real and virtual ones). Therefore the nodes representing such locations are sources. The nodes that represent safety areas are sinks in the final network model. Nodes standing for hallway segments (real and virtual ones) can be interpreted as intermediate nodes. Therefore the set nodes *N* of the evacuation problem can be partitioned as follows (see Chapter 2):

$$N = S \cup I \cup D$$

The different subsets are combining nodes, which represent the same functional segments of a building. Besides a reduction of the network size, the application of aggregation should also result in a reasonable representation of the building. Therefore, it makes sense to state the following assumption for the aggregation of elements of the subsets *S*, *I* and *D* (cp. (AZ1) of Zipkin).

(ASP 1) $\forall J_n \in \overline{NP}$: J_n consist entirely of sources, destinations or intermediate nodes

This means that we only aggregate nodes representing the same functional segments of the building. The assumption should avoid that we lose too much of the problem's specific characteristics due to the aggregation of the network representation. Therefore, we will also assume that nodes representing safety areas will not be aggregated. Since only 18 nodes can be found which represent safety areas in the final model, we would not gain that much network reduction by aggregating one of them. However, leaving them unaggregated leads to a higher level of detail.

(ASP 2) $\forall i \in D : \nexists J_n \in \overline{NP} : |J_n| > 1 \text{ and } i \in J_n$

The last assumption we made on the aggregation of our real world problem instance has something to do with the similarity of locations represented through nodes and the aggregation of such nodes.

(ASP3) Only nodes representing neighboring locations are grouped together

Neighboring locations are locations which are conterminous to each other. This assumption is reasonable, because we do not want to lose too many details by aggregating the problem. The following figure shows us an example for neighboring locations. As we can see it makes no sense to aggregate the nodes which represent office CE.11 and CE.17 without aggregating the nodes representing offices in between.

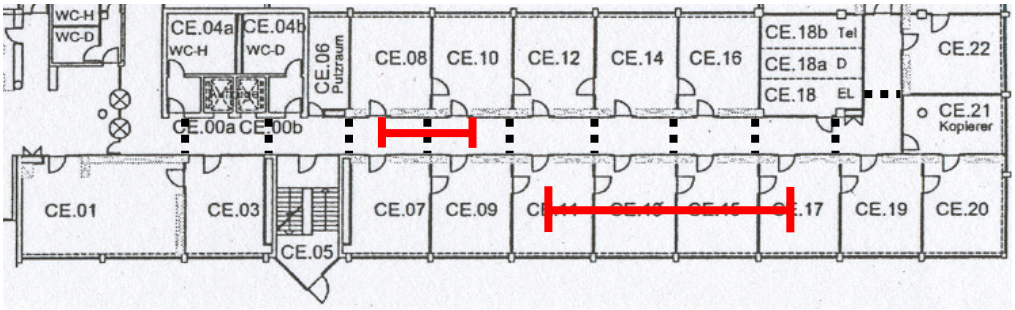


Figure 21: Example for neighboring locations in the blueprint of the EVZ

Note: We will assume that the assumptions (ASP 1-3) are satisfied in the following.

Until now, we have not discussed the application of respecification maps for the vertical aggregation. As mentioned before, it is possible to use the concepts of Chapter 5. However, we will see in the following paragraph that the application of these concepts does not lead to reasonable results. Therefore, we give an own approach for deriving the parameters of the aggregated problem.

6.3.2 Respecification Maps for the Vertical Aggregation

In the last section, we discussed the horizontal and vertical aggregation for dynamic network flow problems. However, only respecification maps for the horizontal aggregation have been defined so far. The definition of them was not complicated, because the maps were predetermined by the characteristics of horizontal aggregation. The vertical aggregation can be compared with the concepts of aggregation discussed in Chapter 4 and Chapter 5 for the transportation problem and the minimum cost network flow problem. The concepts that are discussed there for the costs of an aggregated arc can also be used for the required definition of the travel times of the aggregated evacuation problem. In the aggregation by dominance approach, the costs of an arc $(n, p) \in \overline{A}$ of the aggregated problem are defined as the minimum over all costs of original arcs $(i, j) \in A$, whereby $i \in J_n$ and $j \in J_p$. The capacity for an aggregated arc $(n, p) \in \overline{A}$ is derived by taking the sum over all capacities of original arcs $(i, j) \in A$, whereby $i \in J_n$ and $j \in J_p$. We saw that under this setting the aggregated problem is a relaxation of the original one. It was also possible to derive a bound on the error caused by solving the aggregated problem instead of the original problem setting.
Of course the aggregation by dominance approach can be applied to dynamic network flow problems, too. It is also possible to show that the corresponding aggregated problem is a relaxation of the original problem. We could also take up the idea of the algorithm which derives a sequence of aggregated problems until an optimal solution for the original problem is found. However, from our point of view it is more interesting to get an approximate solution by solving the aggregated problem and derive a bound on the loss of accuracy. Before deriving such a bound for the dynamic case as we have done for the static case (see Proposition 5.2), we should check if it is reasonable to apply this approach to our problem instance, too. In the following example, we made a vertical aggregation for a part of the Office Complex, in which the respecification maps are defined as follows.

Definition 6.1 (Aggregation by Dominance for the Evacuation Problem)

Given an original evacuation problem and a corresponding vertical aggregated evacuation problem based on a partition $\overline{NP}$ of the node set. By using the aggregation by dominance approach the parameters of the aggregated problem can be defined as follows:

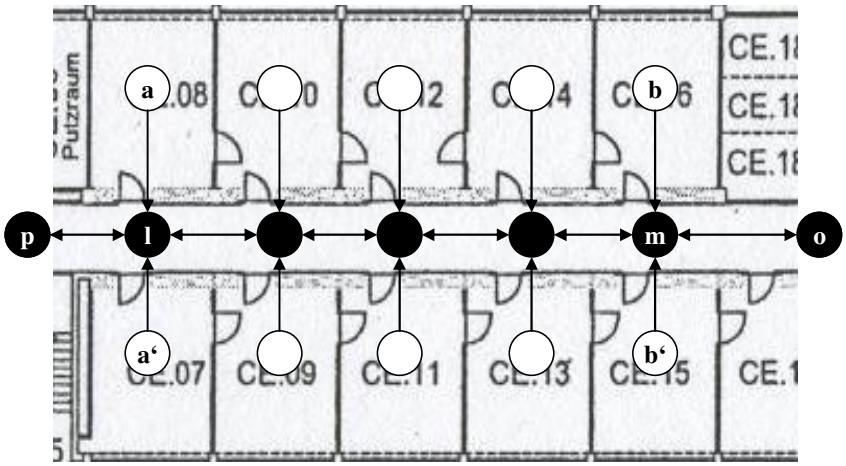
$$\tilde{\lambda}_{np} = \min_{\substack{i \in J_n, j \in J_p: \\ (i,j) \in A}} \lambda_{ij}; \qquad \tilde{u}_{np} = \sum_{\substack{i \in J_n, j \in J_p: \\ (i,j) \in A}} u_{ij};$$

$$\tilde{h}_n = \sum_{i \in J_n} h_i; \qquad \widetilde{EV}_n = \sum_{i \in J_n} EV_i;$$

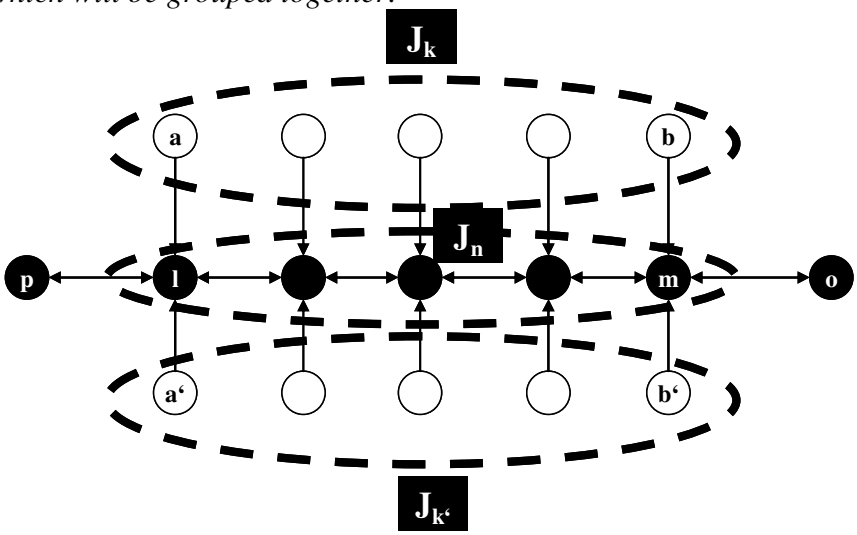
The following example will be used for discussing how reasonable it is to apply the aggregation by dominance approach to the evacuation problem. The example shows a typical aggregation for our problem instance.

Example 6.3

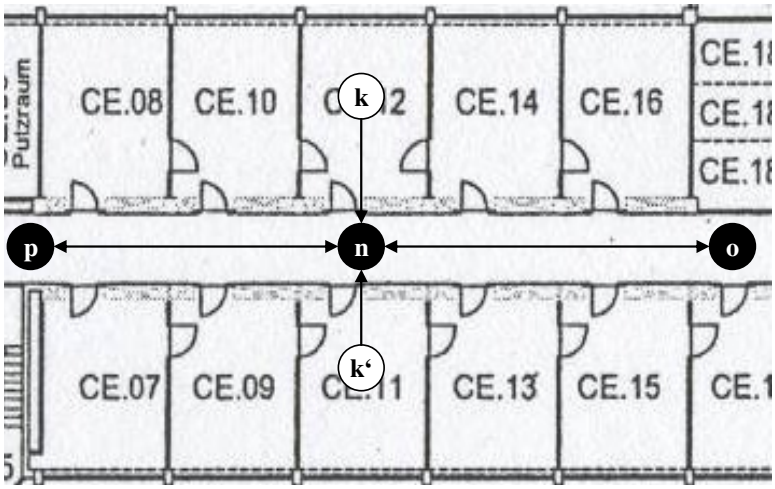
Situation before Aggregation:



Set of nodes which will be grouped together:



Situation after aggregation:



Parameters of the new arcs using the aggregation by dominance approach:

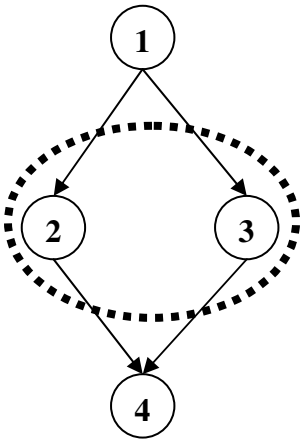
$$\tilde{\lambda}_{no} = \min_{\substack{i \in J_n \\ j \in J_o}} \lambda_{ij} = \lambda_{mo};$$

$$\tilde{\lambda}_{np} = \min_{\substack{i \in J_n \\ j \in J_p}} \lambda_{ij} = \lambda_{lp};$$

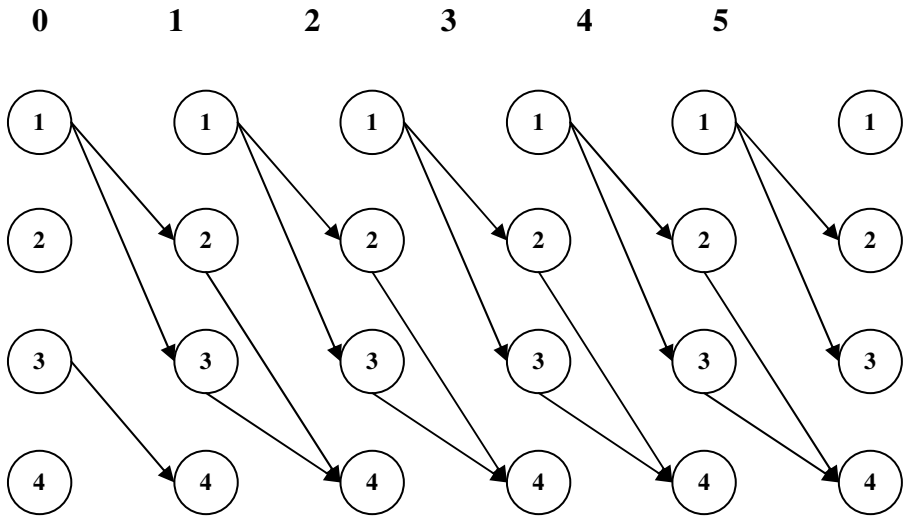
We omit the calculation of the other parameters, because we will focus on the specific characteristics of the travel times calculated above.

As mentioned before, it is obvious that aggregation results in most cases in a loss of accuracy. Our aim is to keep that loss as small as possible while reducing the size of the network. The application of the aggregation by dominance approach leads to a high loss of accuracy for our problem instance. We must take the travel time of (m,o) and (l,p), respectively, for the new arcs (n,p) and (n,o). This means that the evacuees (flow) leaving one of the aggregated rooms k or k' would be directly located on the former hallway segments represented through nodes l and m in the original network. This would be the same as in our original model all evacuees were located in one of the rooms represented through the nodes *a, a', b or b'*. If we aggregate more than two neighboring rooms or hallway segments together, we would lose too much information of the reality. It is clear that in most cases the application of respecification maps will lead to an error. However, the application of the aggregation by dominance approach results in a systematic underestimation of the evacuation time. After all, it makes no sense to apply such an approach to our problem instance.
So let us come to the respecification maps defined in Zipkin's weighted aggregation approach. For the definition of costs in the aggregated problem, Zipkin used some kind of convex combination. In most cases, the costs were not integer. In the same way, his definition of the capacity for aggregated arcs results often in an unfeasible aggregated problem. Nevertheless the main advantage of Zipkin's approach was the fixed-weight disaggregation and the derivation of bounds. However, for the quick disaggregation and the corresponding derivation of the bounds we have to pay a high price. As we have seen in Chapter 5, the aggregation has to satisfy three assumptions, whereof (AZ 2) and (AZ 3) are very restrictive for real world applications. The following example shows that even though the assumptions are satisfied in G_{STA}, the time horizon T and the corresponding holdover possibilities are violating assumption (AZ 3) in the time expanded network.

Example 6.4

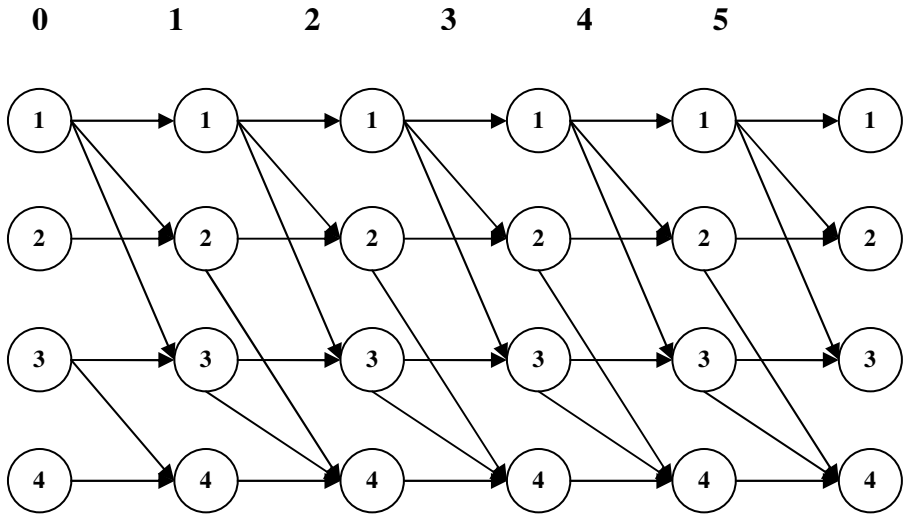


We assume for each arc a travel time equal to one. If we want to aggregate node 2 and 3 the assumptions of Zipkin are satisfied at first view.
If we take a time horizon of five and omit the holdover arcs, we would get the following time expanded representation.



As we can see, the structure of G_{STA} recurs over the time, satisfying the assumptions of Zipkin for all time units $t \in [1,4]$.

If we add the holdover arcs, the third assumption of Zipkin (AZ3) will be violated.



As we see in the example above (AZ3) will be violated by most of the dynamic network flow problems, because of the holdover arcs. Therefore as it is the case for the aggregation by dominance approach, it is not advisable to apply Zipkin's weighted aggregation.

As already mentioned in Chapter 5, it is often necessary to have a more problem dependent derivation of the parameters. Therefore, we should recall one of the basic concepts in the field of modeling evacuation objects. We have seen in Chapter 3 that the nodes representing locations are placed in the middle of this location. This means that the travel time between

two neighboring locations connected through an arc depends on the median distance of both locations. Therefore, it makes more sense to recalculate the parameters of the aggregated problem based on the concepts of Chapter 3 instead of defining an explicit respecification map. Aggregating two nodes is the same as if we consolidate two locations. Hence, the new consolidated location is represented through a node which is placed in the middle of this location. Based on this proceeding, we can calculate the travel time and capacity of the corresponding arcs as well as the holdover capacity and the initial supply of the new node. In the following figures we see this procedure. Node *i* and *j* represent two locations which are connected to other locations. If we group both nodes together, we get the situation of the figure on the right hand side. The new consolidated location is represented through node *n* located in the middle of the new location. Therefore, the travel time from node *n* to other locations and vice versa depends now on the median distance to node *n*. The capacity has also to be adapted depending on the concepts of Chapter 3 and the given situation. To get the supply and holdover capacity of node *n* we can simply sum up the supply and holdover capacity of node *i* and node *j*.

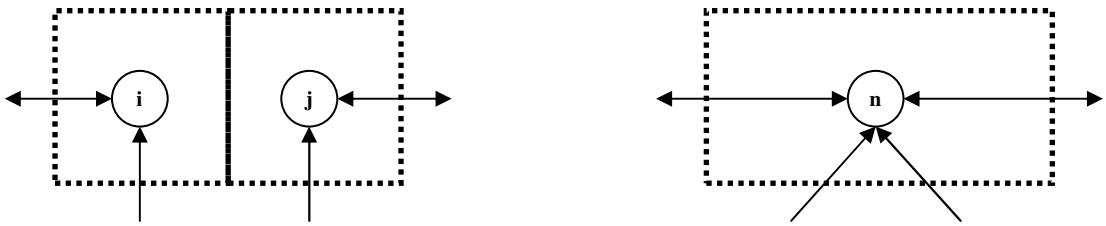


Figure 22: Situation before and after aggregating two nodes

We can derive a formal mapping for the node capacity and the initial supply which holds for all cases of aggregation applied to the evacuation problem satisfying the stated assumptions (ASP1-3).

Definition 6.2

Given an original evacuation problem and a corresponding vertical aggregated evacuation problem based on a partition $\overline{NP}$ of the node set. The holdover capacity as well as the supply for the nodes in the aggregated problem can be calculated as follows:

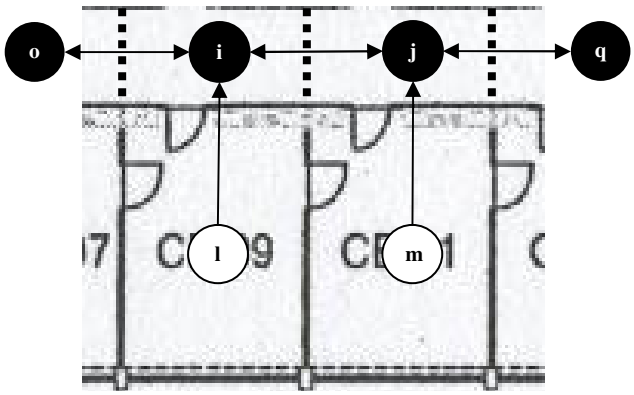
$$\tilde{h}_n = \sum_{i \in J_n} h_i$$

$$\widetilde{EV}_n = \sum_{i \in J_n} EV_i$$

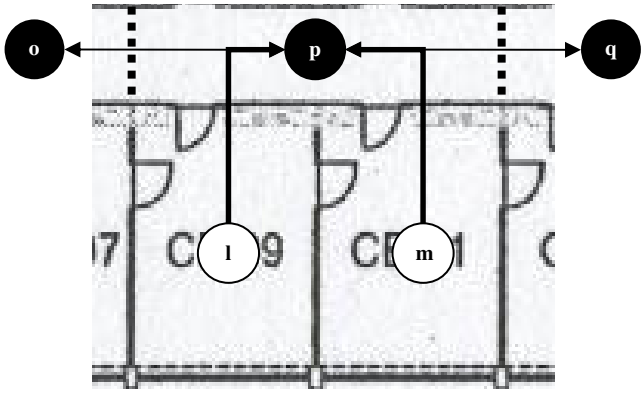
This kind of mapping is obvious and straightforward and uses the data that is already available from the original problem. As mentioned before we take the concepts of Chapter 3 for deriving the travel time and capacity of aggregated arcs. Therefore, we again have to determine distances, lengths, widths etc. . Of course, it is possible to use the original data for recalculating the required parameters. However, the following example shows that it makes no sense to give a formal description for the calculation based on the original parameters. The reason for this is that we have to distinguish different cases of aggregation depending on the problem instance, which result in different calculations of the parameter for the aggregated problem.

Example 6.5

Initial Situation:

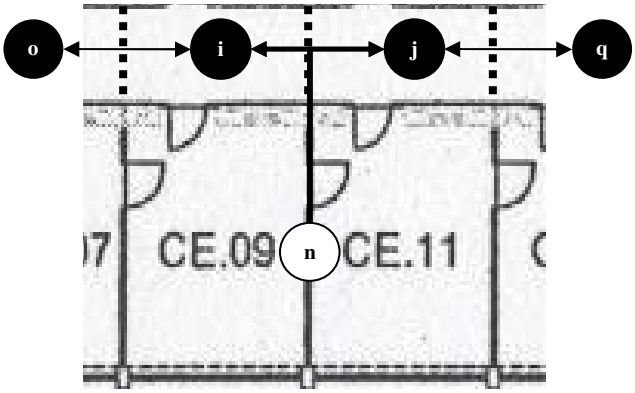


Case 1:



The travel times of arc (l, p) and (m, p) have to be recalculated, because the median distance to the (aggregated) hallway segment has changed. The capacity for the arcs remains the same. The travel time and capacity of (p, q) and (p, o) must be recalculated, too. The node capacity for node p can be calculated as defined above.

Case 2:

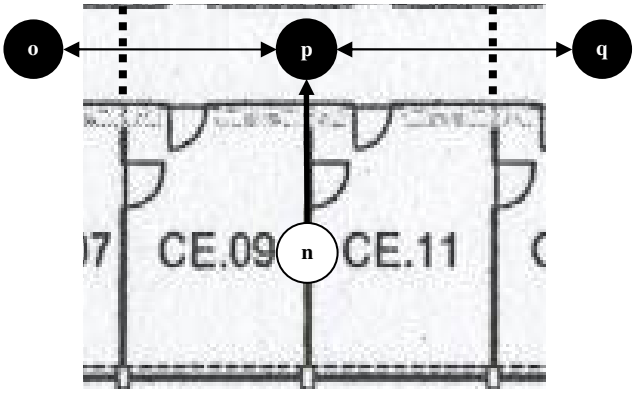


In this case, only the travel times for arc (n,i) and (n,j) have to be recalculated. For the capacity, it holds that:

$$\tilde{u}_{ni} = u_{li} \text{ and } \tilde{u}_{nj} = u_{mj}$$

The node capacity and the initial supply can be calculated with the formula given in Definition 6.3.

Case 3:



The aggregated nodes *p* and *n* represent new locations. This means that the travel time and the capacity have to be recalculated. In this case, the capacity for arc (n,p) can be simply calculated by summing up the original capacity i.e. $\tilde{u}_{np} = u_{li} + u_{mj}$. The node capacity and the initial supply can be calculated as defined above.

Remark: In Example 6.5 we show only an extract of possible case.

Our approach has two main advantages. Firstly, it is conform to our defined concepts of modeling evacuation objects. Secondly, there will be no systematic underestimation of the evacuation time as we have seen when applying the aggregation by dominance approach.

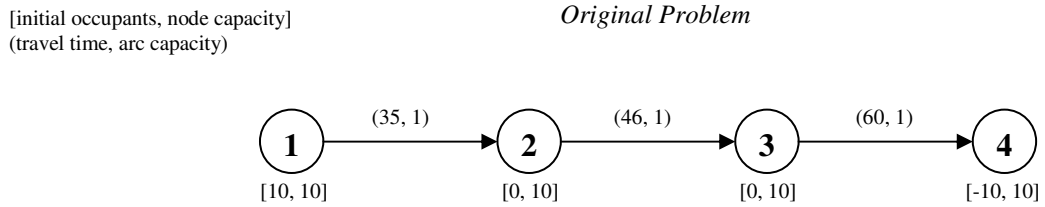
6.4 Loss of Accuracy

In the following section we will examine the loss of accuracy introduced by the horizontal and vertical aggregation. We have seen in Chapter 4 that it was possible to derive two a priori and two a posteriori bounds on the error introduced by aggregating the transportation problem. In Chapter 5 the definition of the minimum cost network flow problem was more general than the one of the transportation problem. The result of this was that we could only derive two a posteriori bounds. However, all the results derived in Chapter 4 and 5 depend on the special characteristics of the respecification maps. In this section we will see that it was not possible for us to derive such a bound for the evacuation problem, when applying our approach for recalculating the parameters of the aggregated problem, saw in the previous section. However, for a special case of aggregation we will be able to derive a theoretical result for the loss of accuracy. Besides this theoretical result, we also want to provide in the following some impressions about the error introduced by the horizontal and vertical aggregation of the evacuation problem.

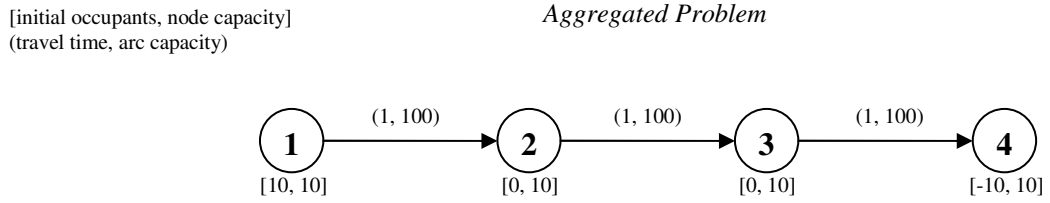
6.4.1 Loss of Accuracy introduced by the Horizontal Aggregation

We assumed that in the original evacuation problem all travel times are measured in seconds. Hence, the basic time unit π is equal to one. In the case of the horizontal aggregation, we just change the basic time unit in order to reduce the number of time copies of nodes and arcs in the time expanded network. To derive an evacuation time for the original problem (i.e. a solution in seconds), the optimal evacuation time of the aggregated problem is multiplied by π. This is a well known procedure for reducing the network size. However, in most cases the consequences of changing π in terms of the solution to the original problem are neglected. If we recall the definition of the respecification map for the travel time, it is obvious that inaccurateness is introduced by rounding the division of the original travel time by the basic time unit.
The following example shows that it is important to take this inaccurateness into account.

Example 6.6



The optimal evacuation time for the original problem is $T^* = 150\,\text{sec} = 2.5\,\text{min}$.
If we set the basic time unit $\pi = 100$ then we get the following aggregated problem



The optimal evacuation time for the aggregated problem is $\overline{T}^* = 3\,\text{time units}$.
We have to multiply the solution of the aggregated problem by the basic time unit π, in order to get an evacuation time for the original problem in seconds.

$$\tilde{T} = \overline{T}^* \pi = 3 * 100 = 300\,\text{sec} = 5\,\text{min}$$

$$\Rightarrow \text{Loss of accuracy} = \left| T^* - \tilde{T} \right| = \left| 2.5 - 5 \right| = 2.5\,\text{min}$$

To minimize the error introduced by the vertical aggregation, it is advisable to set π equal to the greatest common divisor of all travel times whenever possible. Unfortunately, in most cases the greatest common divisor is equal to one. So it is necessary to round up some results. In such a case, it is reasonable to experiment with different choices of π using a small prototype problem.

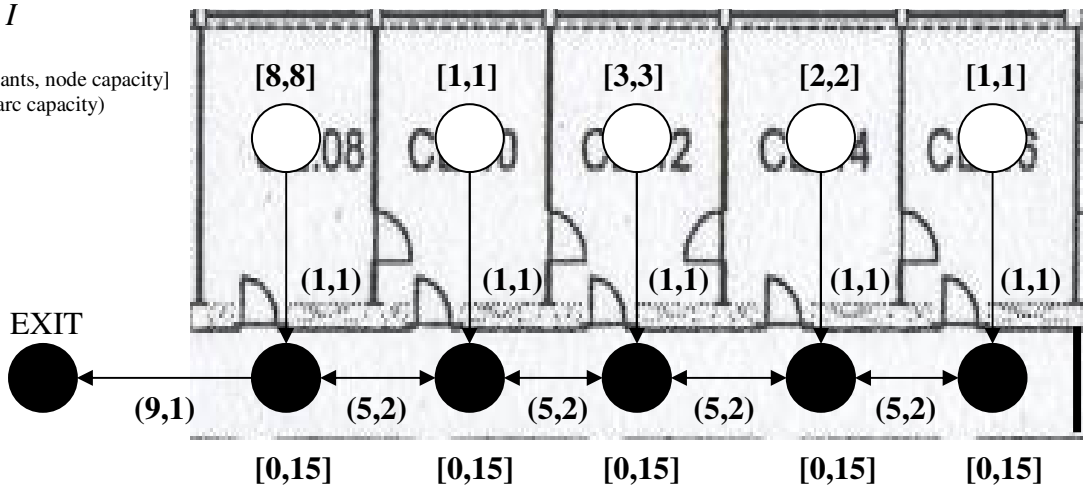
6.4.2 Loss of Accuracy introduced by the Vertical Aggregation

The kind of aggregation applied to the transportation problem and the minimum cost network flow problem in Chapter 4 and 5, is nearly the same as the vertical aggregation stated for the evacuation problem. We saw that it was possible to calculate bounds on the loss of accuracy in the case of the transportation problem and the minimum cost network flow problem. In order to derive the bounds, basic duality theory was used. It was also necessary to know if the optimal objective value of the aggregated problem leads to an upper or a lower bound on the objective value of the original problem, in order to derive a meaningful bound on the loss of accuracy. We have observed in Chapter 5 that the application of the aggregation by dominance approach results in a relaxation. Hence, the optimal objective value of the aggregated problem is smaller or equal than the optimal objective value of the original problem. In contrast to the aggregation by dominance approach used by Lee, the weighted aggregation of Zipkin results in an optimal objective value of the aggregated problem which is greater or equal than the optimal solution of the original problem. As we have seen in the previous section neither the application of the aggregation by dominance approach nor the application of Zipkin's weighted aggregation makes sense for the evacuation problem. From our point of view, it is more reasonable to recalculate the travel time and the capacity based on the concepts of modeling evacuation objects (see Chapter 3). Therefore, we did not define formal respecification maps for the vertical aggregation as it was done by Lee or Zipkin. Nevertheless, we should examine if our approach provides an upper or lower bound on the optimal objective value of the original evacuation problem. It is obvious that we have to give a formal specification for our mapping, if we want to provide a formal proof for any kind of theoretical results for our approach. However, this would be possible but very intricate, since we have to distinguish between several cases of aggregation (see Example 6.5). Therefore, we omit the formal definition so far, because it would rather lead to a confusion of the reader instead of providing some additional value. We will see in the following examples that it is not possible to derive general results on the error introduced by the vertical aggregation. The main problem is that for some instances the optimal evacuation time of the aggregated problem is greater than the one of the original problem and for other instances the opposite holds. Example 6.7 shows a typical problem instance of the evacuation problem. In the first instance of the problem, the EXIT is located on the left hand side, whereas in the second instance it is located on the right hand side. For both instances, we have computed the optimal evacuation time. Then we computed also the optimal evacuation time for the aggregated problem, which is for both instances the same and compare the different values.

Example 6.7

Instance I

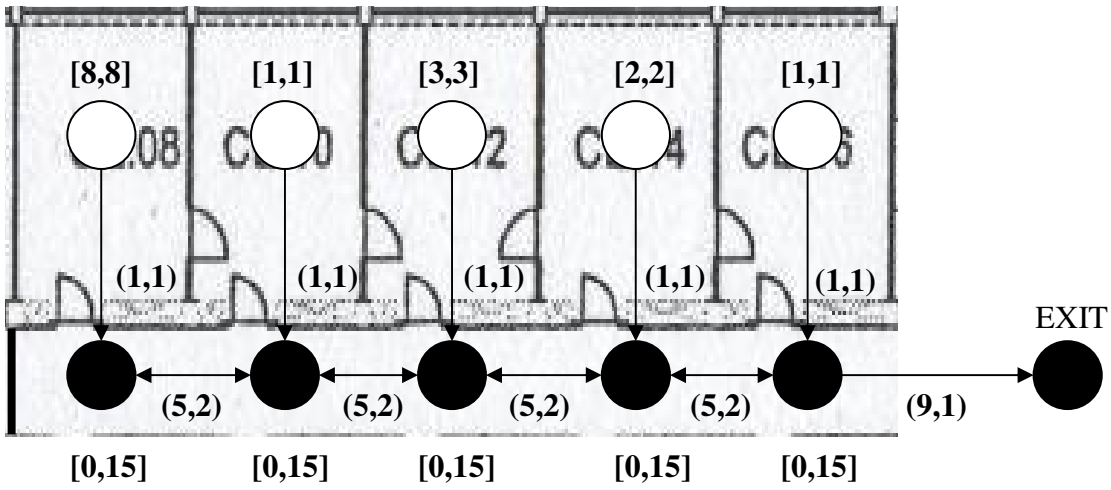
[Initial occupants, node capacity]
(travel time, arc capacity)



The optimal evacuation time for *Instance I* is $T_I^* = 30$

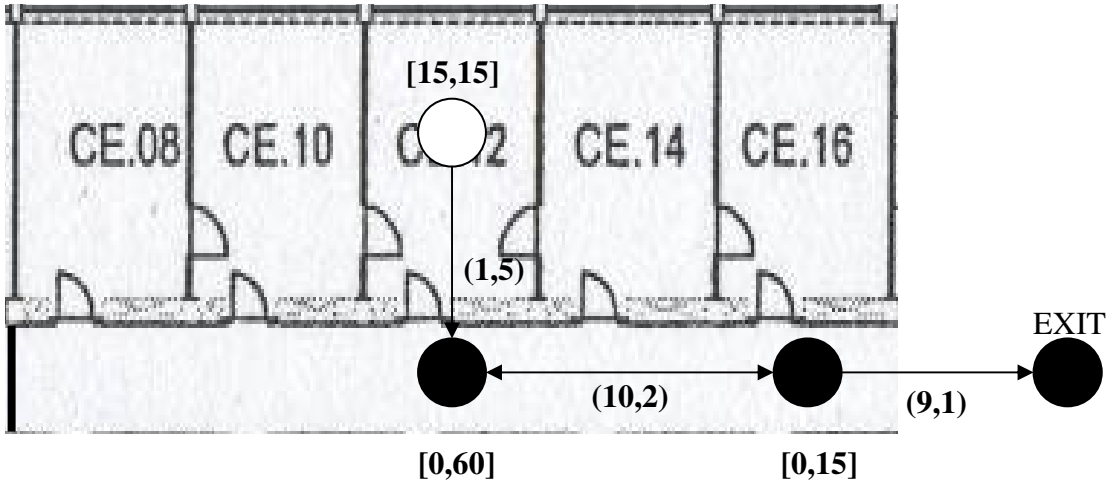
Instance II

[Initial occupants, node capacity]
(travel time, arc capacity)



The optimal evacuation time for *Instance II* is $T_{II}^* = 37$

Corresponding aggregated problem



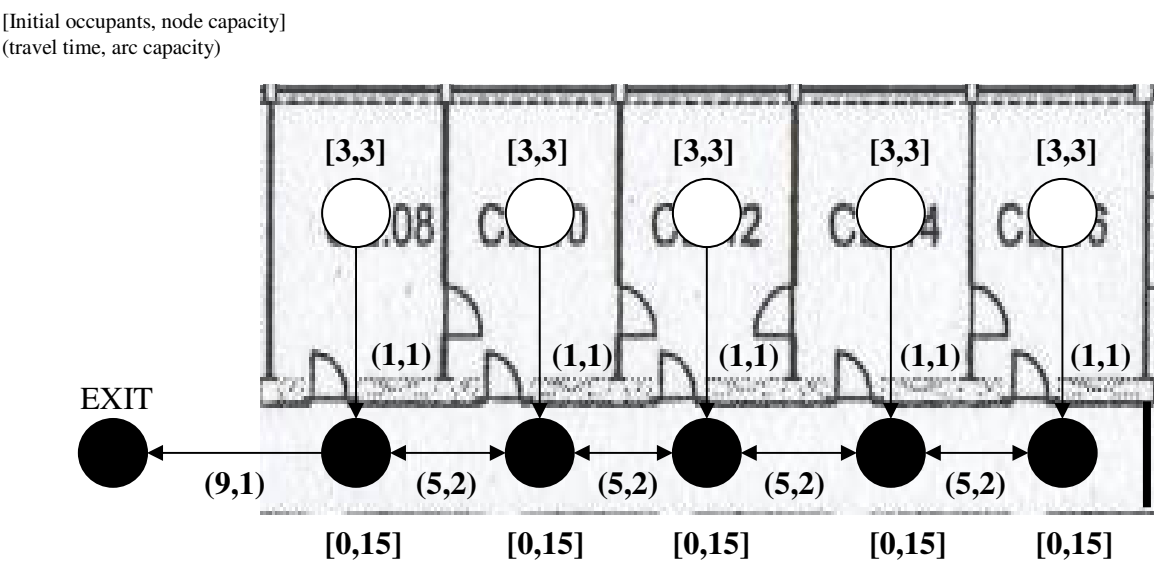
The optimal evacuation time for the *aggregated problem* is $\overline{T}^* = 34$

In the example above, we can see that it will not be possible to show $T^* \leq \overline{T}^*$ or $T^* \geq \overline{T}^*$ in general for our kind of recalculating the parameters of the aggregated problem. The results depend on the distribution of evacuees and the location of the exits. In the following example, we can see that the travel time also plays an important role for the results of the corresponding aggregation. For this example we assume that the evacuees are uniformly distributed over the different offices (i.e. three evacuees per office). In the first instance we used the same travel times as in Example 6.7 resulting in $T^* \leq \overline{T}^*$. If we double the travel time on the hall way

segments, as we have done in it Instance II, we will see that $\overline{T}^* \leq T^*$ holds. Again, it is not possible to derive a general result for the interrelation between the optimal evacuation time of the original problem and the one of the aggregated problem.

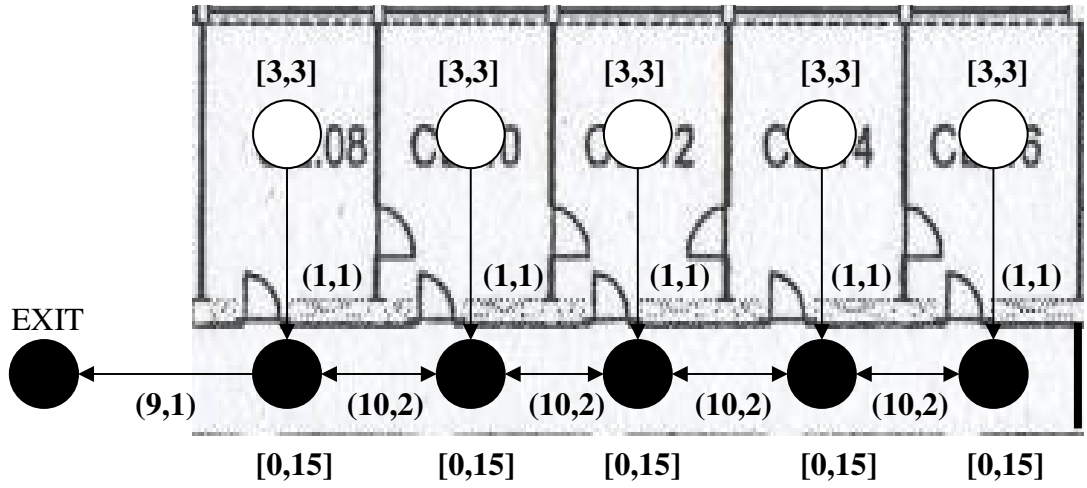
Example 6.8

Instance I



The optimal evacuation time for *Instance I* is $T_I^* = 32$

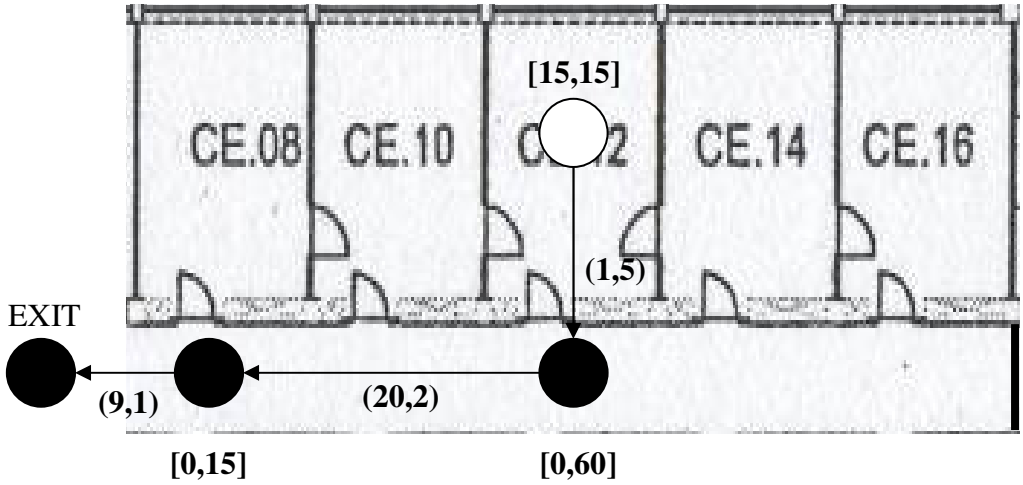
Instance II



The optimal evacuation time for *Instance II* is $T_{II}^* = 52$

The aggregated problem for *Instance I* is the same as in Example 6.7.

Aggregated Problem for Instance II:



The optimal evacuation time for the *aggregated problem of Instance II* is $\overline{T}^*_{II} = 44$

We have seen in Chapter 5 that Zipkin introduced assumptions for the aggregation to force that $T^* \leq \overline{T}^*$. We also tried to come to reasonable assumptions which are not too restrictive in order to get such an inequality, but we failed. So we tried to find some special characteristics in our problem instance especially in the network representation of the Office Complex. In this case, it is easy to identify recurring patterns. This means that if it would be possible to derive some theoretical results for our problem instance of the evacuation problem, we should find it there. Our presumption was confirmed. If we recall the blueprint and the corresponding network representation, we see that whenever possible two opposite offices share one common hallway segment (see the following figure)

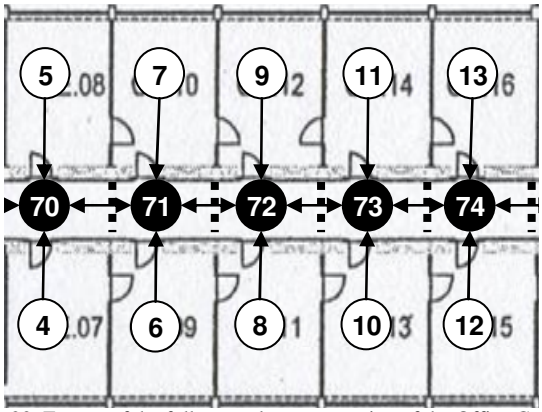


Figure 23: Extract of the full network representation of the Office Complex

The travel time and the capacity of arcs connecting two opposite sources (e.g. 5 and 4) to the same transshipment node (e.g. 70) is the same. We have also assumed that the initial occupancy of the offices is equal (e.g. three occupants in each room). We can show that $T^* = \overline{T}^*$, when only opposite sources are we aggregated together (e.g. 5 and 4) and the respecification maps defined in Theorem 6.1 are applied. This means that instead of solving the original problem, it is sufficient to solve the aggregated problem. Of course, this relationship does not hold in general. It depends on the assumptions made so far and summarized in Theorem 6.1 in a more formal way. The following figure shows the original and the aggregated problem, which are equivalent in terms of the optimal evacuation time.

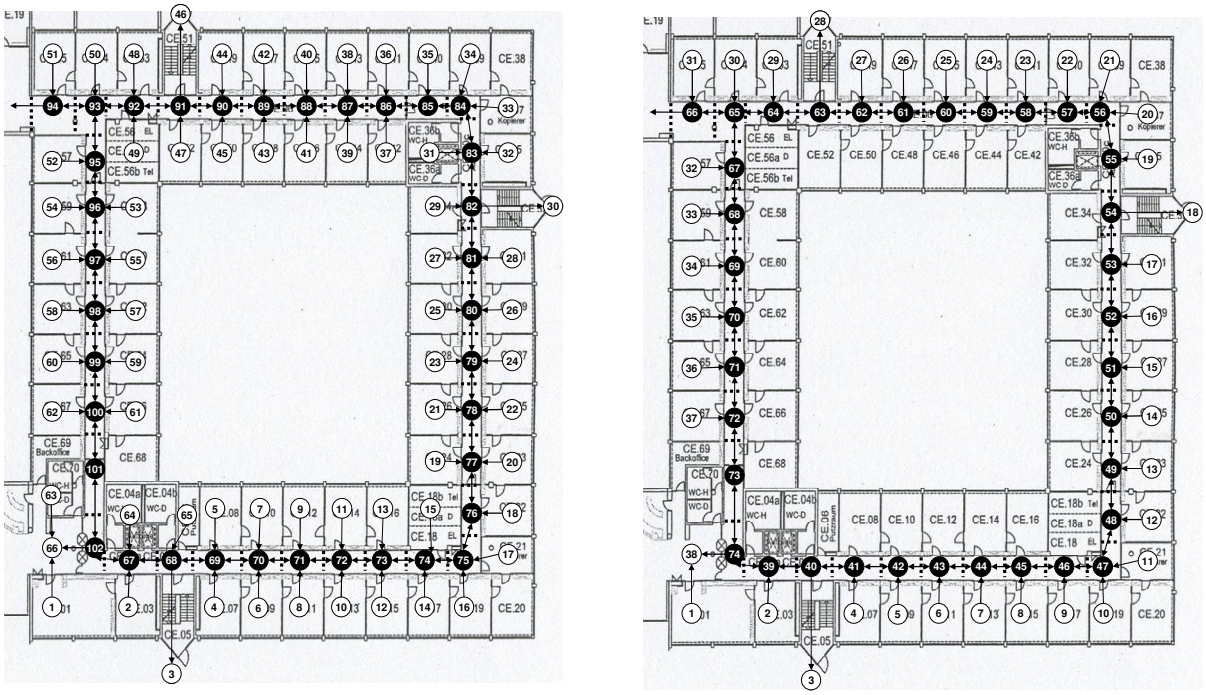


Figure 24: Original and equivalent aggregated representation of the Office Complex

Theorem 6.1

Given is an original evacuation problem with optimal evacuation time T^*. Let VAEVAC be a corresponding vertical aggregated problem, based on a partition $\overline{NP}$ of the node set *N*, with optimal evacuation time $\overline{T}^*$. For $\overline{NP}$ it holds that $|J_q| \leq 2 \quad \forall J_q \in \overline{NP}$ (i.e. at most two nodes of the original problem are grouped together, to a new node in the aggregated problem). In the case of $J_q = \{l, m\}$ (i.e. $|J_q| = 2$), the nodes *l* and *m* satisfy the following assumptions:

- $l, m \in S$
- $\text{pred}(l) = \text{pred}(m) = \{s\}$ (where *s* is the super source)
- $\text{succ}(l) = \text{succ}(m) = \{k\}$
- $k \in I$
- $EV_l = EV_m$
- $\lambda_{lk} = \lambda_{mk}$
- $u_{lk} = u_{mk}$
- $h_l = h_m$

Let us further assume that the node $q \in \overline{N}$ representing *l* and *m* in the aggregated problem has parameters defined by the following respecification maps:

- $\widetilde{EV}_q = EV_l + EV_m$
- $\tilde{\lambda}_{qk} = \lambda_{lk} = \lambda_{mk}$
- $\tilde{u}_{qk} = u_{lk} + u_{mk}$
- $\tilde{h}_q = h_l + h_m$

Node $k \in \overline{N}$ is the same node as in the original problem.

The parameters of nodes and corresponding arcs which left unaggregated are the same as in the original network.

Then it holds that:

$$T^* = \overline{T}^*$$

i.e. the optimal evacuation time of the aggregated problem is equal to the optimal evacuation time of the original problem.

Proof:

Let *l* and *m* be the only two nodes of the original network satisfying the assumptions stated above and grouped together to node $q \in \overline{N}$ in the aggregated problem. The grouping of node *l* and *m* is the only aggregation operation made in the original network.

$$\Rightarrow \quad \overline{N} = \left(N \setminus \{l, m\}\right) \cup \{q\}$$
$$\overline{A} = \left(A \setminus \{(s,l), (s,m), (l,k), (m,k)\}\right) \cup \{(s,q), (q,k)\}$$

With the parameters for *q* and *(q,k)* as defined above.

Claim I $\quad \overline{T}^* \le T^*$

Proof of Claim I:

Let x^* be the flow corresponding to the optimal evacuation time T^*

Define:

$$\overline{y}_{np}(t) = x^*_{ij}(t) \qquad \forall (n,p) \in \overline{A} \setminus \{(s,q),(q,k)\} \text{ where } i \in J_n \text{ and } j \in J_p;\ t = 0,1,...,T;$$

$$\overline{y}_{nn}(t) = x^*_{ii}(t) \qquad \forall n \in \overline{N} \setminus \{q\} \text{ where } i \in J_n;\ t = 0,1,...,T;$$

$$\overline{y}_{sq}(0) = x^*_{sl}(0) + x^*_{sm}(0)$$

$$\overline{y}_{qk}(t) = x^*_{lk}(t) + x^*_{mk}(t) \qquad t = 0,1,...,T;$$

$$\overline{y}_{qq}(t) = x^*_{ll}(t) + x^*_{mm}(t) \qquad t = 0,1,...,T;$$

Proposition 6.1

$\overline{y}$ is a feasible flow for the aggregated problem

Proof of Proposition 6.1:

We have only to proof the feasibility of the flow passing node *q* and *k*, because $\overline{y}$ is equal to an optimal feasible flow x^* of the original problem on arcs, which are corresponding to unaggregated nodes (i.e. $\forall(n,p) \in \overline{A} \setminus \{(s,q),(q,k)\}$). The following figures may by helpful in order to get a better impression why it is sufficient to show that the flow reaching node *k* at each time unit *t* in the aggregated problem is the same as in the original problem.

Original Problem:

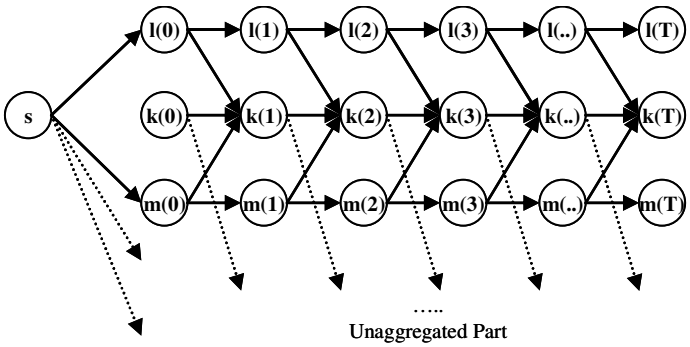


Figure 25: Time expanded network before the vertical aggregation is applied

Aggregated Problem:

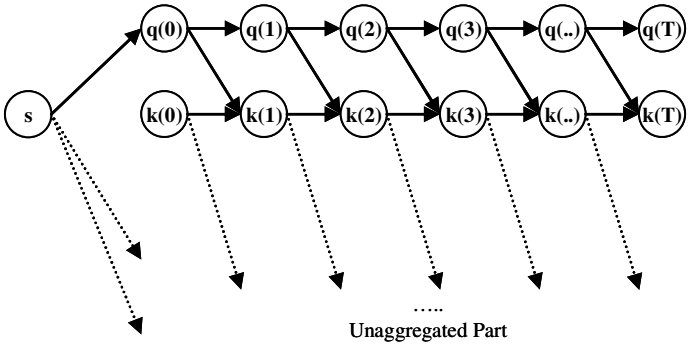


Figure 26: Time expanded network after aggregating node *l* and *m*

We defined the flow $\overline{y}$ in such a way that the same amount of flow as in the original problem reaches node *k* for every time period *t*. After the flow reaches node *k*, the flow distribution is the same as in the original problem. However, it has to be shown that we defined a feasible flow, therefore let us continue with the proof

To show: i.) $\overline{y}_{nn}(t-1) - \overline{y}_{nn}(t) = \sum_{p:(n,p)\in\bar{A}} \overline{y}_{np}(t) - \sum_{p:(p,n)\in\bar{A}} \overline{y}_{pn}(t-\tilde{\lambda}_{pn})$

for $n = q,k;\ t = 0,1,...T;$

ii.) $\overline{y}_{sq}(0) = \widetilde{EV}_q$

iii.) $0 \le \overline{y}_{qq}(t) \le \tilde{h}_q \qquad t = 0,1,...,T$

iv.) $0 \le \overline{y}_{qk}(t) \le \tilde{u}_{qk} \qquad t = 0,1,...,T-\tilde{\lambda}_{qk}$

To i.)

For $n = q$

$t = 0$

$$0 - \overline{y}_{qq}(0) = -x^*_{ll}(0) - x^*_{mm}(0) = x^*_{lk}(0) - x^*_{sl}(0) + x^*_{mk}(0) - x^*_{sm}(0)$$

$$= x^*_{lk}(0) + x^*_{mk}(0) - (x^*_{sl}(0) + x^*_{sm}(0))$$

$$= \overline{y}_{qk}(0) - \overline{y}_{sq}(0)$$

$t > 0$

$$\overline{y}_{qq}(t-1) - \overline{y}_{qq}(t) = x^*_{ll}(t-1) + x^*_{mm}(t-1) - (x^*_{ll}(t) + x^*_{mm}(t))$$

$$= x^*_{ll}(t-1) - x^*_{ll}(t) + x^*_{mm}(t-1) - x^*_{mm}(t)$$

$$= x^*_{lk}(t) + x^*_{mk}(t)$$

$$= \overline{y}_{qk}(t)$$

For $n = k$

$$\overline{y}_{kk}(t-1) - \overline{y}_{kk}(t) = x^*_{kk}(t-1) - x^*_{kk}(t) = \sum_{j:(k,j)\in A} x^*_{kj}(t) - \sum_{j:(j,k)\in A} x^*_{jk}(t-\lambda_{jk})$$

$$= \sum_{j:(k,j)\in A} x^*_{kj}(t) - \sum_{\substack{j:(j,k)\in A \\ l\neq j\neq m}} x^*_{jk}(t-\lambda_{jk}) - \left(x^*_{lk}(t-\lambda_{lk}) + x^*_{mk}(t-\lambda_{mk})\right)$$

$$= \sum_{p:(k,p)\in\bar{A}} \overline{y}_{kp}(t) - \sum_{\substack{p:(p,k)\in\bar{A} \\ p\neq q}} \overline{y}_{pk}(t-\tilde{\lambda}_{pk}) - \left(\overline{y}_{qk}(t-\tilde{\lambda}_{qk})\right)$$

$$= \sum_{p:(k,p)\in\bar{A}} \overline{y}_{kp}(t) - \sum_{p:(p,k)\in\bar{A}} \overline{y}_{pk}(t-\tilde{\lambda}_{pk})$$

To ii.)

$$\overline{y}_{sq}(0) \;= x^*_{sl}(0) + x^*_{sm}(0) = EV_l + EV_m = \widetilde{EV}_q$$

To iii.)

$$0 \overset{*}{\leq} \overline{y}_{qq}(t) = x^*_{ll}(t) + x^*_{mm}(t) \leq h_l + h_m = \tilde{h}_q \qquad t = 0,1,...,T$$

To iv.)

$$0 \overset{*}{\leq} \overline{y}_{qk}(t) = x^*_{lk}(t) + x^*_{mk}(t) \leq u_{lk} + u_{mk} = \tilde{u}_{qk} \qquad t = 0,1,...,T - \tilde{\lambda}_{qk}$$

* holds since x^* is a feasible flow, hence $x^*_{ij} \geq 0 \; \forall (i,j) \in A$

$\Rightarrow \overline{y}$ is feasible for the aggregated problem

q.e.d. (Prop. 6.1)

$\Rightarrow$ Claim 2 holds, since $\overline{y}$ is a feasible flow for the aggregated problem and the evacuation time corresponding to $\overline{y}$ is equal T^*

q.e.d. (Claim I)

Claim II $T^* \leq \overline{T}^*$

Proof of Claim II:

Let $\overline{y}^*$ be the flow corresponding to the optimal evacuation time $\overline{T}^*$ of the agg. problem.

Define:

$$x_{ij}(t) = \overline{y}^*_{n(i)p(j)}(t) \qquad \forall (i,j) \in A \setminus \{(s,l),(s,m),(l,k),(m,k)\};\; t = 0,1,...,T;$$

$$x_{ii}(t) = \overline{y}^*_{n(i)n(i)}(t) \qquad \forall i \in N \setminus \{l,m\};\; t = 0,1,...,T;$$

$$x_{sl}(0) = \frac{1}{2}\overline{y}^*_{sq}(0)$$

$$x_{sm}(0) = \frac{1}{2}\overline{y}^*_{sq}(0)$$

$$x_{lk}(t) = \frac{1}{2}\overline{y}^*_{qk}(t) \qquad t = 0,1,...,T;$$

$$x_{mk}(t) = \frac{1}{2}\overline{y}^*_{qk}(t) \qquad t = 0,1,...,T;$$

$$x_{ll}(t) = \frac{1}{2}\overline{y}^{*}_{qq}(t) \qquad t = 0,1,...,T;$$

$$x_{mm}(t) = \frac{1}{2}\overline{y}^{*}_{qq}(t) \qquad t = 0,1,...,T;$$

Proposition 6.2:

x defined above is feasible for the original problem

Proof of Proposition 6.2:

Since *x* is equal to the optimal flow for the aggregated problem $\forall (i,j) \in A \setminus \{(s,l),(s,m),(l,k),(m,k)\}$, we only have to proof the feasibility of the flow passing nodes *l*, *m* and *k*. (see again Figure 25 and 26 as well as argumentation of the proof for Proposition 6.1)

To show:

i.) $x_{ii}(t-1) - x_{ii}(t) = \sum_{j:(i,j)\in A} x_{ij}(t) - \sum_{j:(j,i)\in A} x_{ji}(t-\lambda_{ji});\ i = l,m,k;\ t = 0,1,...,T$

ii.) $x_{si}(0) = EV_i, \qquad i = l,m$

iii.) $0 \le x_{ii}(t) \le h_i, \qquad t = 0,1,...,T;\ i = l,m$

iv.) $0 \le x_{ij}(t) \le u_{ij}, \qquad t = 0,1,...,T-\lambda_{ij};\ (i,j) \in \{(l,k),(m,k)\}$

To i.)

For $i = l$

$t = 0$

$$0 - x_{ll}(0) = -\frac{1}{2}\overline{y}^{*}_{qq}(0) = \frac{1}{2}(\overline{y}^{*}_{qk}(0) - \overline{y}^{*}_{sq}(0))$$

$$= \frac{1}{2}\overline{y}^{*}_{qk}(0) - \frac{1}{2}\overline{y}^{*}_{sq}(0)$$

$$= x_{lk}(0) - x_{sl}(0)$$

$t > 0$

$$x_{ll}(t-1) - x_{ll}(t) = \frac{1}{2}\overline{y}^{*}_{qq}(t-1) - \frac{1}{2}\overline{y}^{*}_{qq}(t)$$

$$= \frac{1}{2}\left(\overline{y}^{*}_{qq}(t-1) - \overline{y}^{*}_{qq}(t)\right)$$

$$= \frac{1}{2}\left(\overline{y}^{*}_{qk}(t)\right)$$

$$= x_{lk}(t)$$

For $i = m$

See proof for $i = 1$

For $i = k$

$$x_{kk}(t-1) - x_{kk}(t) = \overline{y}^*_{kk}(t-1) - \overline{y}^*_{kk}(t)$$

$$= \sum_{p:(k,p)\in \bar{A}} \overline{y}^*_{kp}(t) - \sum_{p:(p,k)\in \bar{A}} \overline{y}^*_{pk}(t-\tilde{\lambda}_{pk})$$

$$= \sum_{p:(k,p)\in \bar{A}} \overline{y}^*_{kp}(t) - \sum_{\substack{p:(p,k)\in \bar{A} \\ p\neq q}} \overline{y}^*_{pk}(t-\tilde{\lambda}_{pk}) - \overline{y}^*_{qk}(t-\tilde{\lambda}_{qk})$$

$$= \sum_{p:(k,p)\in \bar{A}} \overline{y}^*_{kp}(t) - \sum_{\substack{p:(p,k)\in \bar{A} \\ p\neq q}} \overline{y}^*_{pk}(t-\tilde{\lambda}_{pk}) - \underbrace{\left(\frac{1}{2}\overline{y}^*_{qk}(t-\tilde{\lambda}_{qk}) + \frac{1}{2}\overline{y}^*_{qk}(t-\tilde{\lambda}_{qk})\right)}_{=x_{lk}(t-\lambda_{lk})+x_{mk}(t-\lambda_{mk})}$$

$$= \sum_{j:(k,j)\in A} x_{kj}(t) - \sum_{j:(j,k)\in A} x_{jk}(t-\lambda_{jk})$$

To ii.)

$$x_{sl}(0) = \frac{1}{2}\overline{y}^*_{sq}(0) = \frac{1}{2}\widetilde{EV}_q = \frac{1}{2}\underbrace{(EV_l + EV_m)}_{=2EV_l} = EV_l$$

(same proceeding for $m \in N$)

To iii.)

$$0 \overset{*}{\leq} x_{ll}(t) = \frac{1}{2}\overline{y}^*_{qq}(t) \leq \frac{1}{2}\tilde{h}_q = \frac{1}{2}\underbrace{(h_l + h_m)}_{=2h_l} = h_l \qquad t = 0,1,...,T$$

(same proceeding for $m \in N$)

To iv.) $$0 \overset{*}{\leq} x_{lk}(t) = \frac{1}{2}\overline{y}^*_{qk}(t) \leq \frac{1}{2}\tilde{u}_{qk} = \frac{1}{2}\underbrace{(u_{lk} + u_{mk})}_{=2u_{lk}} = u_{lk} \qquad t = 0,1,...,T-\lambda_{ij};$$

(same proceeding for $(m,k) \in A$)

* holds since $\overline{y}^*$ is a feasible flow, hence $\overline{y}^*_{np} \geq 0 \; \forall (n,p) \in \bar{A}$

q.e.d. (Prop.6.2)

$\Rightarrow$ Claim 2 holds, since the x is a feasible flow for the original problem and the evacuation time corresponding to x is equal $\overline{T}^*$.

q.e.d. (Claim II)

(Claim I and II) $\Rightarrow T^* = \overline{T}^*$

q.e.d. (Theorem 6.1)

Remark: We proofed the theorem in the case there are only two nodes that satisfy the assumptions. However, applying the steps of the proof iteratively, it also holds for the case if more than two nodes satisfying the assumptions are grouped together.

Theorem 6.1 was the only theoretical part that we were able to derive regarding the loss of accuracy induced by the aggregation of the evacuation problem. We have seen in Example 6.8 that the loss of accuracy depends, besides the distribution of evacuees, also on the travel time. To get a better impression what a loss of accuracy we have to expect for our problem instance of the evacuation problem we made in the following some empirical tests for different steps of aggregation applied to the Office Complex.

6.5 Empirical Tests on the Impact of Aggregation

We have seen in the previous section that it was not possible for us to derive general bounds on the loss of accuracy. This was mainly caused by our approach for recalculating the travel times of the aggregated problem. Of course, it would be possible to use the concepts of Lee or Zipkin for the definition of respecifcation maps (i.e. aggregation by dominance and weighted aggregation). The application of their maps would lead to the derivation of bounds. However, as we have seen before, the application of these concepts is not reasonable for our problem instance.
In the following, we will derive some empirical tests on the impact of our recalculation approach. We are aware of the fact that empirical tests are not a real alternative for theoretical results, but we think they can provide an impression of the impact which is caused by aggregation for our problem instance. Therefore, we applied some tests concerning aggregation on the network representation of the Office Complex. We derived five different levels of aggregation for the Office Complex. *Level I* represents the original network, whereas in *Level V* at most five neighboring nodes of the original problem are grouped together to one node in the aggregated problem. The parameters for the aggregated network are computed as defined in Paragraph 6.3.2. The following figures show the different levels of aggregation. The results of Theorem 6.1 are already applied.

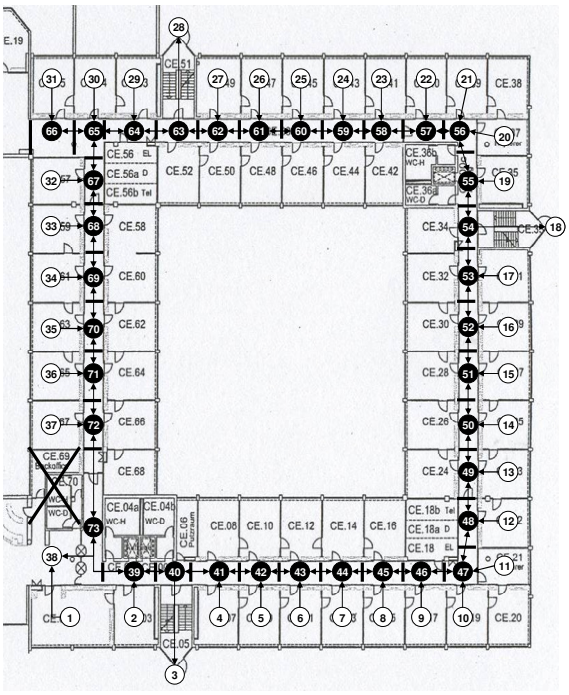


Figure 27: Level of aggregation I for the Office Complex

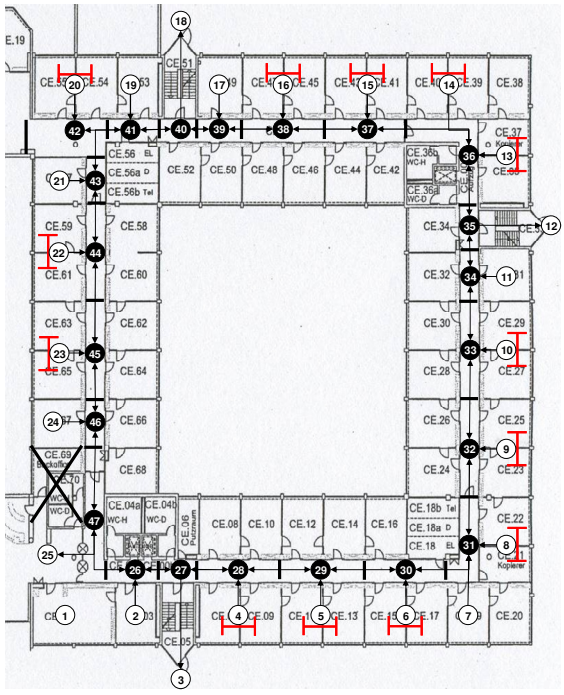


Figure 28: Level of aggregation II for the Office Complex

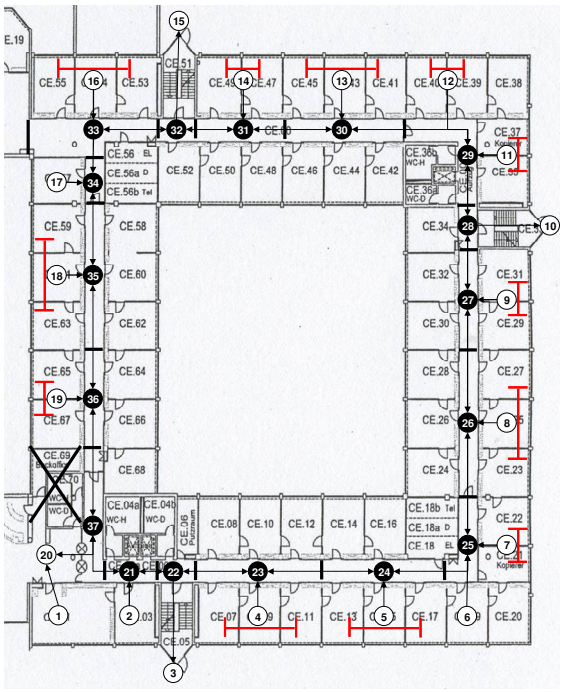


Figure 29: Level of aggregation III for the Office Complex

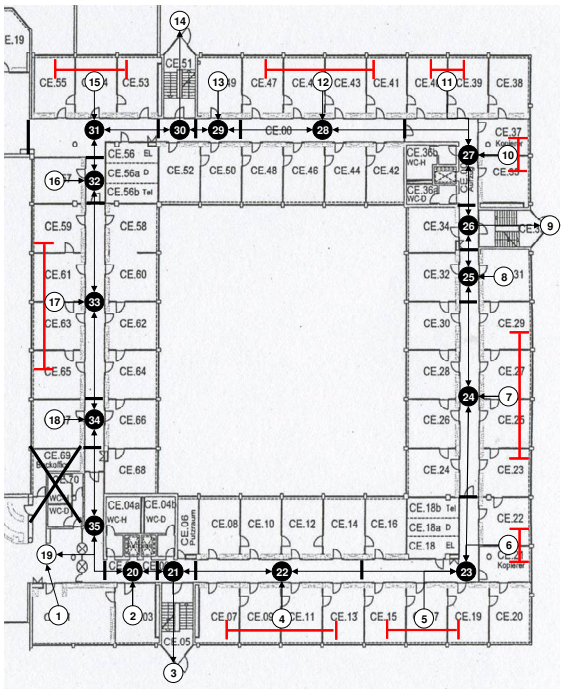


Figure 30: Level of aggregation IV for the Office Complex

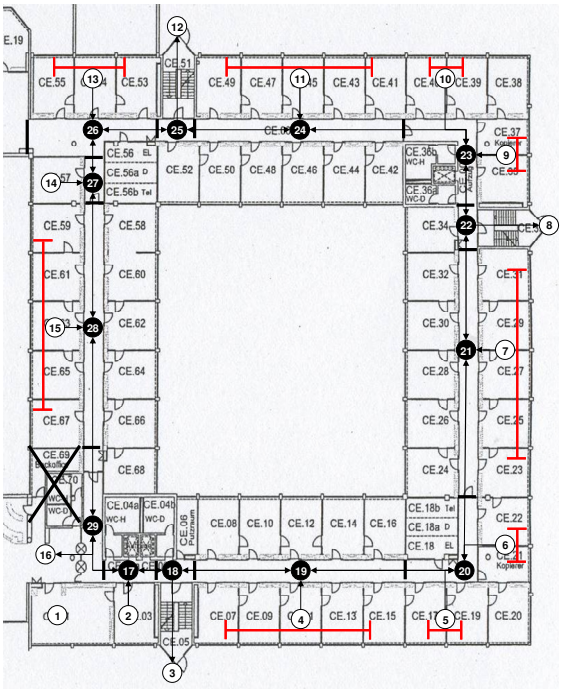


Figure 31: Level of aggregation V for the Office Complex

The tests should give information about the impact of aggregation on the evacuation time. We decided to derive four test instances. In the first instance, it is possible to use all emergency exits modeled in Chapter 3, whereas in the second instance it will not be possible to use the sally port directed to the lobby. In the third instance, not only the sally port but the emergency door of CE.05 as well can not be passed. In instance four, it is only possible to leave the Office Complex via the emergency exit located in the staircase CE.51. The following table provides an overview about the evacuation times for the different test instances.

First Instance

	Evacuation Time	Difference to Level I
Level of Agg. I	*61 sec*	-
Level of Agg. II	*62 sec*	*+ 1 sec*
Level of Agg. III	*64 sec*	*+ 3 sec*
Level of Agg. IV	*64 sec*	*+ 3 sec*
Level of Agg. V	*68 sec*	*+ 7 sec*

Second Instance

	Evacuation Time	Difference to Level I
Level of Agg. I	*73 sec*	-
Level of Agg. II	*74 sec*	*+ 1 sec*
Level of Agg. III	*76 sec*	*+ 3 sec*
Level of Agg. IV	*74 sec*	*+ 1 sec*
Level of Agg. V	*79 sec*	*+ 6 sec*

Third Instance

	Evacuation Time	Difference to Level I
Level of Agg. I	*101 sec*	-
Level of Agg. II	*102 sec*	*+ 1 sec*
Level of Agg. III	*105 sec*	*+ 4 sec*
Level of Agg. IV	*103 sec*	*+ 2 sec*
Level of Agg. V	*109 sec*	*+ 8 sec*

Fourth Instance

	Evacuation Time	Difference to Level I
Level of Agg. I	*186 sec*	-
Level of Agg. II	*187 sec*	*+ 1 sec*
Level of Agg. III	*190 sec*	*+ 4 sec*
Level of Agg. IV	*188 sec*	*+ 2 sec*
Level of Agg. V	*194 sec*	*+ 8 sec*

Table IX: Results of the empirical tests on the impact of aggregation

Remark: For calculating the evacuation times we used an algorithm which was derived in the Ph. D. thesis of Stevanus Tjandra [Tja03] for the earliest arrival flow problem. The implementation of this algorithm and further ones can be found in *LoDyFa* – Library of Dynamic Network Flow Algorithms. To be able to use the algorithm's implementation for the evacuation problem, we have changed the in- and output procedures. An executable version of the algorithm as well as the input files for the test instances can be found on the enclosed CD-ROM. See *http://www.mathematik.uni-kl.de/~wwwopt/evacuation/project.html* for more information about LoDyFa.

In the first instance, we tested the different levels of aggregation on the original network (i.e. all the emergency exits can be used). We see that even though we have in *Level V* a reduction of about *40 %* for the number of nodes, the evacuation time only differs about *11 %*. Because of the fact that the remaining floors of the EVZ have nearly the same structure as the Office Complex, this is an encouraging result, when talking about the application of aggregation. However, we may not forget that there are further factors which must be taken into consideration regarding the modeling of the whole building and the effects of aggregation (e.g. modeling of staircases). In the other test instances, the absolute difference of the optimal evacuation time of *Level I* and *Level V* remains nearly constant. We think that this is mainly caused by our uniform distribution of evacuees over the network. However, the percentages decrease from *8 %* in the second instance to *4 %* in the fourth instance. This effect is caused by congestion effects in front of the different emergency exits. The less emergency exits are opened the more congestion occurs.
In our tests we do not examine different levels for the basic time unit π. Of course this is also an important parameter regarding aggregation of evacuation problems. However, we think that paragraph 6.4.1 provides enough information about the impact of this kind of aggregation.

Chapter 7

Conclusion and Outlook

The first part of this thesis discussed the modeling of evacuation processes by using discrete-time dynamic network flow problems. In the second part, which is also the main section, we concentrated on aggregation theory and its application to the evacuation problem.
We reviewed the evacuation problem in detail and gave an overview about further dynamic network flow problems which can be used for modeling evacuation processes. We discussed the modeling of buildings in general and had an application to a real world problem instance in particular. The concepts of Fruin for modeling the parameters of an evacuation process were reviewed. Based on the network representation of a building, we showed how the evacuation problem can be solved. However, we saw that the network size is the main problem of solving the evacuation problem. Therefore, we reviewed in a next step aggregation theory applied to static network flow problems such as the transportation problem and the minimum cost network flow problem. We saw that the approaches found in literature could be divided into two parts, depending on the provided kind of solution for the original problem (optimal vs. feasible solution). These two approaches can be named by the respecification maps they use. The *aggregation by dominance*, introduced by Balas, leads to an aggregated problem that is a relaxation of the original one. The concepts which are based on the aggregation by dominance approach focused on the derivation of algorithms, which finally lead to an optimal solution for the original problem. In the case that no optimal solution is needed, we derived a bound for the loss of accuracy caused by solving the aggregated problem instead of the original one, when the aggregation by dominance approach is applied.
The *weighted aggregation* introduced by Zipkin allows a very easy derivation of a feasible solution for the original problem, based on a solution for the aggregated one. Hence, the application of this approach finally leads to an upper bound on the optimal objective value of the original problem. Even though Zipkin's concepts have some appealing properties (e.g. weighted disaggregation), we have pointed out that their application to real world problems is very restrictive. After having reviewed the theory for static problems, we tried to transfer the observations made there to the evacuation problem. In that case, two dimensions regarding aggregation could be identified. In the *horizontal aggregation* the basic time unit π is changed. By applying this kind of aggregation, we pointed out that whenever possible and reasonable to set the basic time unit equal to the greatest common divisor of all travel times. The *vertical aggregation* is nearly the same kind of aggregation we already saw for the transportation problem or the minimum cost network flow problem. However, it was not reasonable to apply the respecification maps defined for these problems to our instance of the evacuation problem, too. Therefore, we defined a different approach for recalculating the parameters. By using this approach, the optimal solution of the aggregated problems leads neither to an upper bound nor to a lower bound on the optimal objective value of the evacuation problem. Therefore, it was not possible for us to derive a bound on the loss of accuracy for the evacuation problem. Nevertheless, we showed a theoretical result for a special case of aggregation in which the optimal evacuation time of the aggregated problem is the same as the optimal evacuation time of the original one. The empirical tests concerning aggregation we made in the end of Chapter 6 encouraged us to assume that the impact of

aggregation is not that bad for the evacuation problem. However, the empirical results indicated that we have to apply the sandwich approach, as seen in Chapter 1 (Figure 2), carefully when using aggregation. We assumed for this approach that the *macroscopic model*, represented by the evacuation problem in our thesis, should yield a lower bound on the evacuation time. But when we applied aggregation to our problem instance, we saw that the evacuation time of the aggregated problems was greater than the one of the original ones. This means that it has to be examined if the optimal solution for aggregated evacuation problems is still a lower bound for the evacuation time. Of course, we also saw in examples that the optimal solution of the aggregated problem may be smaller than the one of the original problem. In such cases, the sandwich approach would still hold. We should again stress the fact that this approach has to be validated, but it is important to take the impact of aggregation for this validation into account.

We conclude this thesis with a discussion about some future research topics related to the modeling of evacuation processes and the application of aggregation to dynamic network flow problems.

Modeling

In this thesis, we assumed a constant crowd level for modeling the travel time and capacity of arcs. The application of flow dependent parameters would increase the model quality. Besides flow dependent parameters, the application of multicriteria objective functions would enhance the possibilities for modeling evacuation processes, too. These are well known approaches for modeling evacuation processes, but the impact on aggregation has not been investigated so far. As a first step, the application of time dependent parameters could be considered in order to get a first impression about this impact.
In the case of modeling a complete building with several floors, staircases have to be taken into account. The modeling of such building parts is interesting for two reasons: on the one hand, the modeling of staircases is a challenging issue mainly depending on the particular problem instance. On the other hand, it would be interesting to observe the impact on aggregation.
All networks modeled in this thesis are modeled by hand. This means that we had to detect distances between all the different locations to model parameters such as the travel time of arcs, for instance. No doubt about the fact that this is an extensive task for a network with over 200 nodes and arcs. All the data we need for the calculation of parameters is already available in CAD files, in which most blueprints are available. Since our aim is to model complete buildings in a reasonable amount of time, it should be investigated how the CAD files can be used to automatize the modeling process.
We saw in the beginning that there exist two models, for mapping evacuation processes; the macro- and microscopic model. We also mentioned the supposed interrelation between them (see again Figure 2). In a future research this interrelation should be examined, by applying both models to a real world instance and comparing the results with data gained from evacuation practices.

Aggregation

We saw that is was not possible for us to derive a reasonable bound for the loss of accuracy for the evacuation problem. This was mainly caused by our definition of recalculating

parameters. The respecification maps of the weighted aggregation or the aggregation by dominance were defined in such a way that a derivation of bounds is possible. However, we saw that their application fails for our problem instance. In a future research, it should be examined how a respecification map can be defined for the evacuation problem which is not only reasonable for the problem instance, but also allows the derivation of a bound.
We only made empirical tests concerning the impact of aggregation for a small part of the EVZ, so far. It would be interesting to know more about the side effects regarding aggregation when the complete building is modeled.

Appendix

Instead of an appendix in written form, we decided to enclose a CD-ROM, with additional information which should facilitate the understanding of the thesis.
After plugging in the CD-ROM, it should directly start with an overview menu, which was programmed in HTML and can be viewed with any known browser. If the CD-ROM does not start automatically you can access the overview by a double click on "*index.htm"*.

Bibliography

[AMO 93] R.K. Ahuja, T.L. Magnati and J.B. Orlin: "Network Flows: Theory, Algorithms and Applications". *Prentice Hall, Englewood Cliffs, New Jersey, 1993*

[Bal65] E. Balas. "Solution of Large-Scale Transportation Problems Through Aggregation". *Operations Research, 13: 82-93, 1965*

[BA99] V.J. Blue and J.L. Adler. "Using Cellular Automata Microsimulation to Model Pedestrian Movements". *Proceedings of the 14th International Symposium on Transportation and Traffic Theory, Jerusalem, Israel, 1999*

[BDK93] R.E. Burkard, K. Dlaska, and B. Klinz. "The Quickest Flow Problem". *ZOR-Methods and Models of Operation Research, 37:31- 58, 1993*

[CFS82] L.G. Chalmet, R.L. Francis and P.B. Saunders. "Network Models for Building Evacuation". *Management Science, 28: 86-105, 1982*

[CS00] M. Carey and E. Subrahmanian. "An Approach To Modelling Time-Varying Flows On Congested Networks". *Transportation Research B, 34: 157-183, 2000*

[EPRW91] J. R. Evans, R.D. Plante, D.F. Rogers and R.T. Wong. "Aggregation and Disaggregation Techniques and Methodology in Optimization". *Operations Research, 39: 553-582, 1991*

[FF58] L.R. Ford and D.R. Fulkerson. "Constructing Maximal Dynamic Flows from Static Flows". *Operation Research, 6: 419-433, 1958*

[FF62] L.R. Ford and D.R. Fulkerson. "Flows in Networks". *Princeton University Press, Princeton, New Jersey, 1962*

[Fra85] J.R.Francis. "Aggregation of Network Flow Problems". *Ph. D. Thesis, University of California, Los Angeles, USA, 1985*

[Fru71] J.J. Fruin. "Pedestrian Planning and Design". *Metropolitan Assiciation of Urban Designers and Environmental Planners, 1971. out of print*

[FT98] L. Fleischer and E. Tardos. "Efficient Continuous-Time Dynamic Flow Algorithms". *Operations Research Letters, 23: 71-80, 1998*

[Gal59] D. Gale. "Transient Flows in Networks". *The Michigan Mathematical Journal, 6: 59-63, 1959*

[HT94] B. Hoppe and E. Tardos. "Polynomial Time Algorithms for some Evacuation Problems". *Proc. Of 5th Annual ACM-SIAM Symposium on Discrete Algorithms, 433-441, 1994*

[HT01] H.W. Hamacher and S.A. Tjandra. “Mathematical Modeling of Evacuation Problems: A State of the Art ”; *Published in M. Schreckenberg and S.D. Sharma “Pedestrian and Evacuation Dynamics”, pages 227-266, Springer 2001*

[JR82] J.J. Jarvis and H.D. Ratliff. “Some Equivalent Objectives for Dynamic Network Flow Problems”. *Management Science, 28: 106-108, 1982*

[JTC95] R. Jayakrishan, W.K. Tsai and A. Chen. “A Dynamic Traffic Assignment Model with Traffic-Flow Relationships”. *Transportation Research C, 3: 51-72, 1995*

[KFN98] T.M. Kisko, R.L. Fancis and C.R. Nobel. “Evacnet4 User’s Guide”. *http://www.ise.ufl.edu/kisko/files/evacnet/EVAC4UG.HTM*, *1998*

[KMSW00] H. Klüpfel, T. Meyer-König, M. Schreckenberg and J.Wahle. Microscopic “Simulation of Evacuation Processes on Passenger Ships”. *Fourth International Conference on Cellular Automata for Research and Industry, October, Karslruhe, Germany, 2000*

[Lee75] S.J. Lee. “Surrogate Programming by Aggregation”. *Ph. D. Thesis, Department of Mathematics, University of California, Los Angeles, USA, 1975*

[Lod05] AG Optimization. “LoDyFa – Library of Dynamic Network Flow Algorithms”, *Department of Mathematics, University of Kaiserslautern, Kaiserslautern, 2005*

[Pau78] J. Pauls. “Evacuation of High Rise Office Buildings”. *Buildings, 5:84-88,1978*

[Pau82] J. Pauls. “Effective-Width Model for Crowd Evacuation Flow on Stairs”. *295-306, Sixth International Fire Protection Seminar, Karlsruhe, Germany, September 21-24, 1982*

[RZ83] K. Raimer and P.H. Zipkin. “An Improved Disaggregation Method for Transportation Problems”. *Mathematical Programming, 26: 238-242, 1983*

[Sti00] K. Still. “Crowd Dynamics”. *Ph. D. Thesis, Department of Mathematics, University of Warwick, Warwick, 2000*

[Tay83] R.W. Taylor. “Aggregate Programming in Large Scale Linear Systems”. *Ph. D. Thesis, Georgia Institute of Technology, Atlanta, USA, 1983*

[Tja03] S.A. Tjandra. “Dynamic Network Optimization with Application to the Evacuation Problem”. *Ph. D. Thesis, Department of Mathematics, TU Kaiserslautern, Kaiserslautern, Germany, 2003*

[Wor05] www.WordReference.com, *2005*

[Zip77] P.H. Zipkin. “Aggregation in Linear Programming”. *Ph. D. Thesis, Yale University, New Haven, USA, 1977*

[Zip80] P.H. Zipkin. “Bounds for Aggregating Nodes in Network Problems”. *Mathematical Programming, 19: 155-177, 1980*

List of Figures

List of Tables

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Referenzen und Hilfsmittel erstellt zu haben. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt.

Pirmasens, August 2005

Florian Dreifus